THIS PAGE BLANK (USPTO)

PCT/GB 99/02125

GB99/02125

S

REC'D 30 JUL 1999
WIPO PCT

THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office

June 1, 1999

THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY FROM THE RECORDS OF THE UNITED STATES PATENT AND TRADEMARK OFFICE OF THOSE PAPERS OF THE BELOW IDENTIFIED PATENT APPLICATION THAT MET THE REQUIREMENTS TO BE GRANTED A FILING DATE UNDER 35 USC 111.

APPLICATION NUMBER: ***60/091,665***
FILING DATE: ***July 2, 1998***

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN COMPLIANCE WITH RULE 17.1(a) OR (b)

PATENT AND TRADEMARK OFFICE · UNITED STATES OF AMERICA

By Authority of the
COMMISSIONER OF PATENTS AND TRADEMARKS

N Williams

N. WILLIAMS
Certifying Officer

Express Mail Label
EE 790841283

A/PROV

July 2, 1998

THE COMMISSIONER OF PATENTS AND TRADEMARKS

Washington, D. C. 20231

Sir:

Transmitted herewith for filing in accordance with 35 U.S.C. 111(b) is the **PROVISIONAL PATENT APPLICATION** of **MARTYN GILBERT**, titled **COMPUTER SYSTEM ARCHITECTURE** together with a cover sheet, a Verified Statement to Establish Small Entity Status under 37 CFR 1.9 and 37 CFR 1.27, and a Check in the sum of $75 for the requiste provisional application filing fee for a small entity.

Respectfully submitted,

Kenneth A. Roddy

Kenneth A. Roddy
Agent for Applicant
Registration No. 31,294
Telephone (713) 686-7676

Encls.

CERTIFICATE OF MAILING - **EXPRESS MAIL**

I hereby certify that this correspondence is being deposited with the United States Postal Service **"EXPRESS MAIL POST OFFICE TO ADDRESSEE"** service under 37 CFR 1.10, postage prepaid, in an envelope addressed to:

Commissioner of Patents and Trademarks,
Washington, D.C. 20231, on July 2, 1998

Date: July 2, 1998

Kenneth A Roddy

Kenneth A. Roddy
Agent for Applicant
Registration No. 31,294

2916 West T.C. Jester Blvd.
Suite 105
Houston, Texas 77018
(713) 686-7676

EXPRESS MAIL LABEL
790841283

PTO/SB/16 (8-96)

Appro[illegible]r use [illegible] 01/31/98. OMB 0651-0037

Patent and Trademark [illegible]; U.S. D[illegible]ARTMENT OF COMMERCE

PROVISIONAL APPLICATION FOR PATENT COVER SHEET

This is a request for filing a PROVISIONAL APPLICATION FOR PATENT under 37 CFR 1.53 (b)(2).

Docket Number		Type a plus sign (+) inside this box →	

INVENTOR(s)/APPLICANT(s)

LAST NAME	FIRST NAME	MIDDLE NAME/INITIAL	RESIDENCE (CITY AND EITHER STATE OR FOREIGN COUNTRY)
Gilbert	Martyn		41 St. Michaels Longstanton, Cambridge England C34 5B2

TITLE OF THE INVENTION (280 characters max)

COMPUTER SYSTEM ARCHITECTURE

CORRESPONDENCE ADDRESS (including country if not United States)

Kenneth A. Roddy
Suite 105
2916 West T.C. Jester
Houston, TX 77018

ENCLOSED APPLICATION PARTS *(check all that apply)*

- [X] Specification — *Number of Pages* 68
- [X] Drawing(s) — ~~Number of Sheets~~ INCLUDED WITH TEXT OF SPECIFICATION
- [X] Small Entity Statement
- [X] Other (specify) CERTIFICATE OF MAILING EXPRESS MAIL

METHOD OF PAYMENT OF FILING FEES FOR THIS PROVISIONAL APPLICATION FOR PATENT *(check one)*

- [X] A check or money order is enclosed to cover the filing fees
- [ ] The Commissioner is hereby authorized to charge filing fees and credit Deposit Account Number:

FILING FEE AMOUNT ($) $75

The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.

- [X] No.
- [ ] Yes, the name of the U.S. Government agency and the Government contract number are:

Respectfully submitted,

SIGNATURE *Kenneth A. Roddy*

Date 07/02/98

TYPED or PRINTED NAME Kenneth A. Roddy

REGISTRATION NO. *(if appropriate)* 31,294

- [ ] Additional inventors are being named on separately numbered sheets attached hereto

USE ONLY FOR FILING A PROVISIONAL APPLICATION FOR PATENT

EXPRESS MAIL LABEL
EE 790841283

VERIFIED STATEMENT CLAIMING SMALL ENTITY STATUS (37 CFR 1.9(f) & 1.27(b)) - - INDEPENDENT INVENTOR

Docket Number Optional)

Applicant or Patentee: MARTYN GILBERT

Application or Patent No.:

Filed or Issued:

Title: COMPUTER SYSTEM ARCHITECTURE

As a below named inventor, I hereby declare that I qualify as an independent inventor as defined in 37 CFR 1.9(c) for purposes of paying reduced fees to the Patent and Trademark Office described in:

[X] the specification filed herewith with title as listed above.

[ ] the application identified above.

[ ] the patent identified above.

I have not assigned, granted, conveyed, or licensed, and am under no obligation under contract or law to assign, grant, or license, any rights in the invention to any person who would not qualify as an independent inventor under 37 CFR 1.9(c) if that person had made the invention, or to any concern which would not qualify as a small business concern under 37 CFR 1.9(d) or a nonprofit organization under 37 CFR 1.9(e).

Each person, concern, or organization to which I have assigned, granted, conveyed, or licensed or am under an obligation under contract or law to assign, grant, convey, or license any rights in the inventon is listed below:

[ ] No such person, concern, or organization exists.

[X] Each such person, concern, or organization is listed below.

AMINO COMMUNICATIONS LTD.

I acknowledge the duty to file, in this application for patent, notification of any change in status resulting in loss of entitlement to small entity status prior to paying, or at the time of paying, the earliest of the issue fee or any maintenance fee due after the date on which status as a small entity is no longer appropriate. (37 CFR 1.28(b))

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under section 1001of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application, any patent issuing thereon, or any patent to which this verified statement is directed.

Martyn Gilbert	Martyn Gilbert	15-6-98
NAME OF INVENTOR	Signature of Inventor	Date
NAME OF INVENTOR	Signature of Inventor	Date
NAME OF INVENTOR	Signature of Inventor	Date

Express Mail Label
EE 790841283

U.S. Provisional Patent Application

COMPUTER SYSTEM ARCHITECTURE

This Provisional Patent Application for the invention titled "Computer System Architecture" is comprised of several sections as follows:

1. The HomeWeb
2. IntAct System Architecture
3. Architectural Modules
4. IntAct Secure Bus Technology
5. IntAct Bus Specification
6. IntAct and Legacy Systems

The drawing figures corresponding to the written description are included in the text portion of each respective section.

HomeWeb - A Discussion Document

Introduction

HomeWeb is the logical extension of present-day domestic and commercial Intranet and Internet technology. It acknowledges that products and technologies designed to satisfy the present notion of one Internet appliance per building or user will prove totally inadequate as the market matures.

The suggestions in this document recognise that the granularity of Internet and Infranet connection will decrease. For example, in the home, consumers will not only use Internet technology with their televisions but children will make extensive use of it for home assignments (homework - GB). These activities may well occur simultaneously although a home may only have a single, shared Internet connection. The need to make a emergency telephone calls at a time that an Internet appliance is autonomously using the domestic telephone connection without blocking the former highlights the need to control the shared resource.

Internet appliances will become smaller - security devices from cameras to smoke alarms offer the opportunity for improved peace of mind at reduced cost. Premises surveillance can be made more effective at lower cost.

All of this needs to be remotely controllable with a high degree of consumer confidence.

In the commercial market, low cost premises surveillance of this type can help reduce insurance premiums or even make the uninsurable an acceptable risk.

In medical applications, the prospect for improved patient monitoring without the need to design custom infrastructures can help improve healthcare.

Some Basic Thoughts

The wide diversity of applications will require that a number of products are engineered. Core products that can deliver a basic, marketable solution for a clearly defined subset of the market will be needed first. After launch, the regular introduction of new products and services will help ensure the concept remains fresh and potential competitors are

'wrong footed'. This approach will also help reduce up-front development funds as revenue will help fund subsequent elements of the system.

System Structure

Partitioning

The preceding descriptions illustrate the acknowledged dilemma between traditional mainframe servers and clients: Just how much of the resources are put in the servers and how much in the clients. The networked computing and protocols issue raised in this document are simply an extension of that discussion. In fact, it is apparent that in a distributed computing and resources environment, there is a continuum of complexity and functionality between service provision and service consumption.

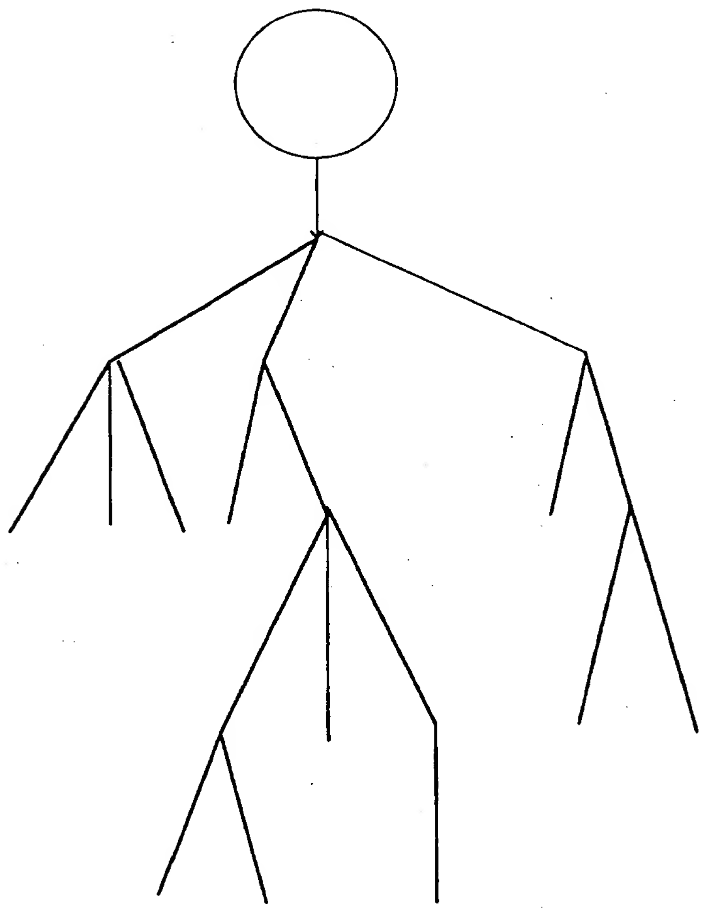
This suggests that even the interconnecting cables in a system will embody some information processing be it active and/or passive. This of course is exactly what they do. Usually, that processing degrades system performance. Properly designed, the effects are beneficial although active attributes are often used to offset information degradation caused by passive degradation.

At this stage in the development of the technology, a true continuum of processing is not practicable. A discrete model is more appropriate. Such a model is a network of connected active and passive information processing nodes. The network is not homogenous because of asymmetry caused by security controls. With respect to a human user, security introduces the bottom-up restricted access. The asymmetry brings order to the network by constraining the possible routes of information flow.

In any given network, the direction of the asymmetry is not necessarily the same for any two applications. For example, a set-top-box (STB) might be part of a security structure for content conditional access. It may also be part of another structure for e-commerce. Conditional access may be determined by the STB's internal mechanisms and a remote server, co-located with the content provider. The e-commerce end-point may be in a key pad held by the user but served by a virtual private network connecting individual elements of e-commerce such as certification, invoicing, value acquisition etc. This could well result in an almost random, formless network. Unless one is prepared to accept a stochastic model of bandwidth distribution, to maintain any degree of

control of quality of service - indeed, reliability - form must be re-introduced to the structure.

The model to be adopted is a traditional hierarchical tree structure, with high level core services at the top and branches of successively lower bandwidth going down. Security of data is controlled from the bottom up.



The server circle at the top can contain any single server or network of servers as might be modelled by a corporate server or the Internet.

At the end of each line in the structure exists some primarily client device that can however offer server facilities. Typically, bandwidth

reduces down the diagram. This reduces the prospect for a geom tric increase in bandwidth requirement as the systems get larger.

It will be readily apparent from the diagram that the structure is similar to the bandwidth structure of PSTN networks, indeed, of many highly ordered switched networks. The lesson here is that Telco's have worked out this structure for the same sorts of reason's over many years. The difference is that the model is now being applied to what is essentially a local area network. LAN's have been dominantly logical shared buses over the last few years, despite the fact that they may be wired as point-to-point physical networks (as in IEEE802.3 10baseT Ethernet). Thus the local area network can become a seamless extension of the wide area network, differing only in the physical implementation and electrical signalling protocol.

Network Protocols

It is highly desirable to feature a single network type running throughout the system to ensure Quality of Service (QoS). This suggests that ATM is the best choice. That would satisfy the objective of seamless integration between the WAN and LAN environments. Only the aggregate bandwidth would vary at a given point in the system. For much of the LAN, this could be 25Mbit, decreasing to 2Mbit and less to handheld's etc. The actual data rate could massively vary within that range. The stated figures are burst rates, based on established electrical signalling protocols.

To do this with ATM is at the time of writing, commercially impractical. Silicon would need to be developed to provide ATM switching within the humblest of devices. For example, A STB could have on-going connections to a handheld, another consumer device (e.g. digital VCR, thus avoiding any analogue links in the system). Most networks would be very small (see the Server and mClient description below.

A lot of installations would require adaptations of ATM to run in a shared medium such as handheld's and lower cost networks[1] in which providing logical switching on a shared bus would be cheaper than physical switching[2].

Software

A wide variety of topologies need to be supported. In general, the system architecture is independent of the operating system used in any

[1] ATM being carried over USB comes to mind.
[2] Come back ARCNET. All is forgiven.

particular system component. However, some operating systems may be inherently unsuitable for larger, more diffuse installations. Conversely, other operating systems may be very well suited to large systems but could be wasteful of resources in smaller examples.

In due course, a new operating system, or adaptation of an existing one may be required. For example, Windows CE could be quite appropriate for single CPU systems, whereas TAOS or HELIOS are designed to support multiprocessor systems.

The choice of operating system is largely a compromise between development effort and cost, time-to-market and technical reliability and determinism. Almost certainly, a dedicated inter-processor protocol will be needed for larger systems. Smaller, multiprocessor, molecular (as opposed to atomic) systems can be built with simple extensions to existing operating systems. Traditional Real Time Operating Systems (RTOSs) are well suited to these.

It is anticipated that the first products to market will use an existing OS with mere device driver extensions.

It is preferable that all system components feature replaceable software so that enhancements can be introduced during the product lifetime.

Security

One of the reasons for multiprocessor systems is to support the notion of data security. This may be to ensure authorised control, e-commerce or just plain privacy.

The system must therefore support hierarchical scope of data structures. Access to a given level is controlled by that level alone. Higher level data users must request lower level sub-systems for their data and be prepared to authenticate their request. Levels that do not wish to guard their data may freely pass requests and responses through themselves. In part, this supports the trend for Java based subsystems.

Level transparency applies to both hardware and software. A higher level hardware access of a lower level system may be permitted to pass through the lower layer unmodified. If blocked, the lower level will make the access on behalf of the upper level, if permitted. This suggests the availability of exception handling facilities within the

processor at a given level[3]. It is not desirable to mandate the use of exception processing elements at all levels in the system due to the affect that would have on cost. Therefore, any CPU that does NOT support exception processing MUST represent an end-point in the hierarchy, unless it NEVER makes hardware requests to a lower level.

A good example of this would be in a simple set-top-box that is controlled via an infra-red remote control equipped with an integral Smart Card interface. Clearly, the set-top-box cannot make a hardware access to the Smart Card and must rely on a software protocol over the optical link. The CPU in the set-top-box need not support exception processing.

Performance

Initially, the protocols between and inside levels must be simple. This may have a detrimental effect on data throughput and care must be taken in the design of the protocols to prevent this.

Implementation Structure

A fully fledged domestic installation is based on the idea of at least one web server in the premises. This provides services to 'micro-clients" (mClients hereafter). These may have very little by way of hardware or software. It is the server that provides a full-bodied Internet presence for mClients. The degree of proxy facilities a server provides will in turn depend on how much resource a given mClient actually needs. For example, serving five separate televisions displaying different video-on-demand images may be impractical. Serving five simultaneous Internet HTML pages would be quite straightforward.

Smaller installations may not have the local server in which case greater computing capability will be required in the mClients which then take on an appearance more akin to that of traditional clients.

Host Networks

This refers to the connection to the outside world. As such it implies WAN connections.

- PSTN (POTS)
- ISDN (Basic and primary rate)
- ADSL

[3] To cause a software protocol to be invoked in lieu of an aborted hardware access.

- Cable Modem
- Power Line modem

These are the most probable types. However, Fixed Link Radio should also be considered as it way prove popular in countries with less developed wired infrastructure.

Whilst cellular telephones are inappropriate for most applications, they may prove useful as a tamper-proof back-up to static installations e.g. if a telephone line is cut. It may also prove useful in mobile applications such as overnight surveillance of long distance haulage trucks and telemetry from same.

The choice of host network will depend on geographical, logistical and application constraints. Typically, they will operate as midband or broadband services for high data rates (>1Mbit/s) or baseband for low data rate (<1Mbit/s).

Local Networks

By definition, these are local area networks and are suited to all data rates.

- Ethernet
- 25Mbit ATM
- Local power line Modem
- 2.4GHz spread spectrum
- Broadband coaxial
- CAN

The term "broadband coaxial" refers to the use of 75 ohm television cable with premises. This is most likely to be installed plant but would also be suitable for a wide variety of low cost, simple function new installations. For example, a central VoD server could transmit an RF modulated service for reception by standard, unmodified televisions without the need for a Set Top Box. Internet and other services could also be provided, albeit with some degradation compared to a baseband service. Channel selection and simple email would use an infra-red keyboard. The IR device would transmit to a small IR receiver in a small box using a mains carrier modem to connect back to the server.

Local power line modem could be a feature-rich technology such as Echelon but may also be a simpler, custom designed system.

Control Area Networks (CAN) may be useful in automotive applications.

System Components

Web Server

The server would be the most costly part of the system.
Simple, monolithic implementations would be needed for low cost applications such as may be installed by builders to add value to a new home without adding much cost.

Larger chassis based, multi-card designs would be appropriate for the enthusiast. Between the two, an low-end chassis solution would be more appropriate for deployment by cable companies and PTTs.

Elements of the server would be

- One or more host network connections (POTS and satellite may be needed simultaneously).
- Main CPU
- ROM (FLASH)
- DRAM
- Conditional access device
- Hard Disc
- One or more local network connections (e.g. mains carrier modem and Ethernet may be needed)
- Zero or more mClient line cards (display controller, screen buffer, colour encoder modulator).

mClients

These small devices would typically be the size of a cigarette packet. They would have power, network and display connections. User I/O may just be infra-red although a range of option would be needed. User smart cards may go into a companion IR keypad/board with a conditional access smart card inside the mClient.

They would be very low cost - especially if mains carrier modems were used in them to connect back to the server.

For single installation premises (e.g. one browser in a home) then the same mClient could use soft modem technology to avoid the cost of the Web server which could not be justified for a single client. The mClient would then need internal display controller electronics.

Modular Construction and Economy of Scale

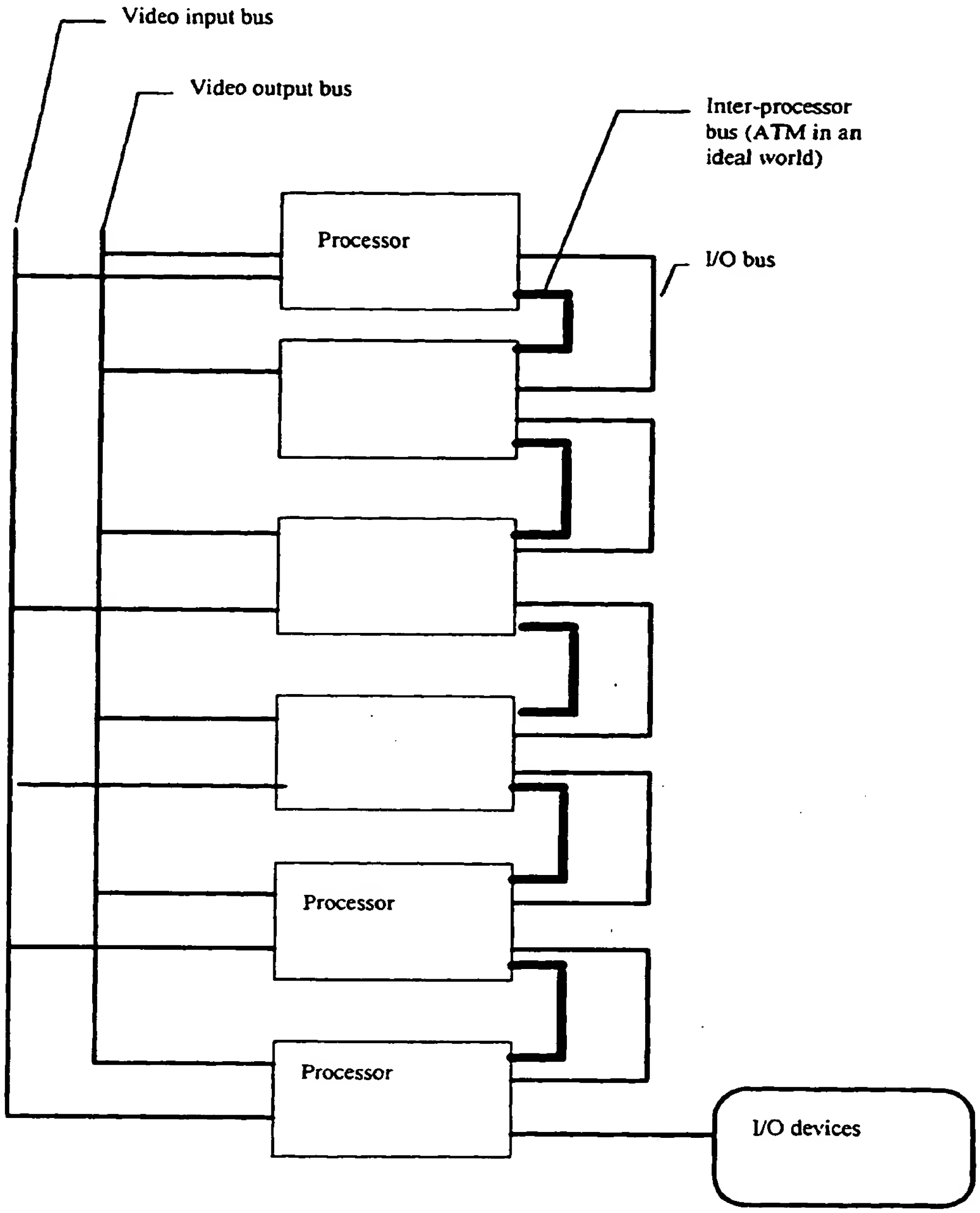
It is most desirable that the structure of components at the micro level reflects that of the macro. This suggests that even a STB in this system is itself an assembly of hierarchical components connected by internal ATM. This would massively reduce system interconnection requirements and would facilitate the security model. This Utopian view sadly has to be tempered by the fact that ATM so intimately integrated into the system is not a commercial reality unless significant engineering investment is made. In the meantime, system interconnect has to be constructed out of more traditional busses. Laboratory demonstrations can of course be constructed.....

By constructing system components (the nodes in the tree structure) out of modules which are hierarchical in their own right, we can fashion either very slender mClients or more substantial network computers of arbitrary complexity.

From a manufacturing standpoint, this retains economy of scale despite that a variety of differing end products are to be made.

A Typical Client Device

We now examine the structure within a client node.



The main feature is that it is shown as a collection of processors. These need not be exclusively arithmetic. They could be audio, video and with their own, intimately bound I/O. Such I/O is deemed to be indistinguishable from the processing that supports it. "Processors" are of arbitrary complexity and need not exist. This permits a device to be built with audio, video, networking etc. provided by a single processor and thus have no security protection; through to a multi-processing, highly secure hierarchical device.

Notice that all busses other than video are potentially gated by higher level processors. This obviously provides the security. A processing node that does not wish to apply any secure controls to a bus can just pass it through, unmodified.

1

IntAct System Architecture

System and Component level topology

Introduction

System Model

IntAct system components fit into a hierarchy:

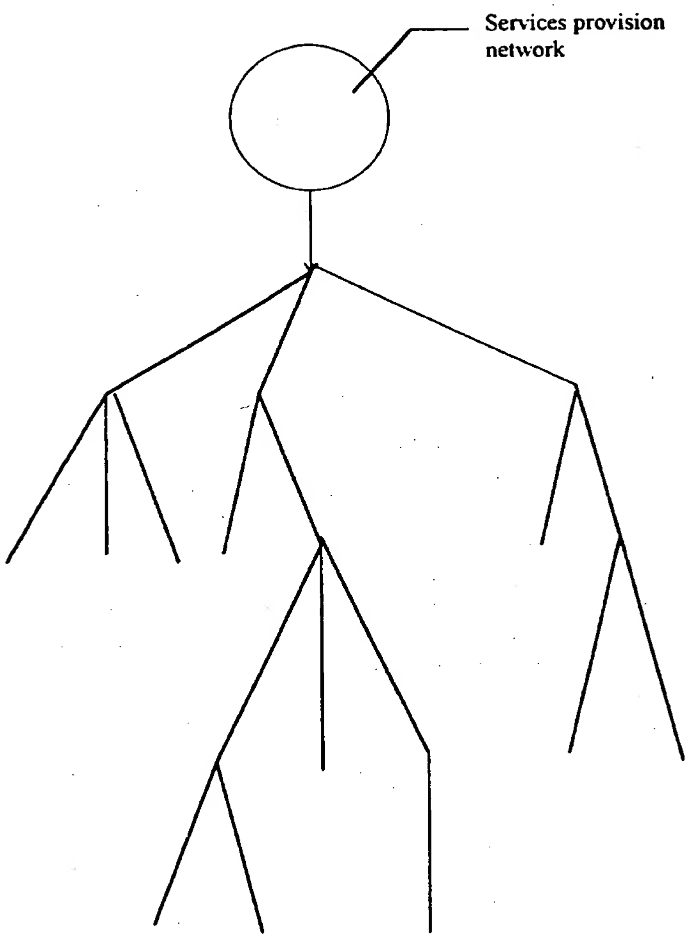


Figure 1 - Hierarchical Structure

At each node in the system, an active information processing device manipulates the data passing though it. The minimum amount of processing may be null; what comes in goes out. Alternatively, very little of the original information may be passed down; a completely different, yet related message may be generated.

This feature allows hierarchical security to be implemented:

- Nodes higher up the structure control provision of services to lower components.
- Nodes from the bottom of the structure control end user authentication.

Essentially, higher nodes facilitate server-side security; lower nodes facilitate client-side security.

Typically, higher nodes consume or distribute larger amounts of system bandwidth than lower nodes. This model of course is the same as public switched telecommunications e.g. dial-up or ISDN.

The model does not put any constraint on network topology. Quality of Service does however *guide* us into making some specific requirements of the system:

- User interface devices shall operate with minimal perceptible delay.
- User authentication and permission devices shall be secure from higher level interference.
- The system shall appear to be 'transparent' to Audio-visual content[1].

Very small, single node IntAct systems may be constructed as could systems with thousands of nodes. The bulk of early implementations will most likely have no more than three nodes. Eventually, systems with up to thirty nodes seem likely.

In the tree structure, any vertical line between nodes represents a security level. Child nodes from a single parent are deemed to be at the same security level. Related nodes from a given grandparent node are however treated as being in different security levels as data between them must travel up to the grandparent and thus traverse two levels and so be open to attack.

Other attributes are:

- The message protocol between nodes is independent of the transport protocol.
- The transport protocol is connection orientated.

[1] It will not be obvious to an end user that such a system is in the datapath compared to a traditional analogue television.

Component Requirements

- Active components at each node follow the broad description of the system level. The purpose is to allow the security structure of the structure to permeate that of the system components.
- The deterministic bandwidth requirements of the system are intended to supply the individual components with data at the rate needed to satisfy the functional specification

Taken together, the above constraints suggest that ATM[2] is the ideal system transport protocol. At the time of writing, commercially available silicon is not available to provide low cost ATM in all system components.

For example, ATM is sometimes used in ADSL and cable modems for the final connection to the client device. ADSL or cable modem end-to-end connections then require ATM at the server or 'head-end' equipment.. This adds complexity when used with conventional Wide Area Networking (WAN) software technologies. Specifically, TCP/IP in the local area requires specialised support (e.g. LANE) to be carried over ATM. The lack of this support will cause difficulties for client-end software that only has TCP/IP support. This difficulty only arises however, if the entire system structure is compliant with industry standard protocols such as those specified by the ATM Forum.

For the above reasons, the systems architecture must allow installations that comply with the hierarchical topology yet must permit non-ATM transport protocols in the near-term. The architecture must permit a migration to full, system-wide ATM.

It is intended that this is provided by assigning the node closest to the services provision as the gateway node between the ATM and non-ATM environments. Usually, this will also correspond to the interface between the WAN and LAN portions of a given installation.

For example, ATM may be used as the final connection from an ADSL modem into a building. Within the building, a TCP/IP over 10baseT Ethernet may be deployed. LANE functions are then only required in the gateway node.

Conversely, 25Mbit ATM may be used within a building but the gateway may only be provided with an ISDN connection to the outside world.

Practical Considerations

- Domestic systems must be feasible using installed cable plant.

This single requirement dictates that multiple data rates must be supported in the local area. It would be Utopian if 25Mbit ATM switches and cable plant were installed in 80% or more of residential premises. Sadly, this is not the case.

The following must be supported:

- Mains Carrier Modem
- Category 5 twin twisted pair
- 75Ω coaxial cable
- Wireless

1. [2] Asynchronous Transfer Mode

- Consumer Infra-Red

Data rates in these physical layers vary over five orders of magnitude. This dictates that the same date rate cannot be used throughout the system. These rates however *do not* dictate that ATM and ATM-like structures cannot be used in the signaling specifications.

This enables us to build systems in the near term which may be upgraded to full ATM standards over a period of time. Initially, they need only employ the principles of ATM.

Component Model

The system structure employing ATM principles is carried into the component level design. That is, components such as servers, set-top-boxes, generalised 'Thin Clients' must be constructed to the same hierarchical model.

The component model described in this document facilitates the same hierarchical structure as the macro system whist acknowledging the near-term economic constraints in building such a system. That is, the component architecture offers ATM-like data propogation whilst facilitating other, lower cost systems.

The structure shown in Figure 2 has a number of processing modules, connected via a communications path and a control path for ancillary peripheral parts. Dedicated video busses prevent user video devices from 'swamping' the main inter-processor bus. The Utopian implementation would just have ATM going from module to module. Consequently, Figure 2 should be viewed as an implementation of the architecture, not a statement of definition.

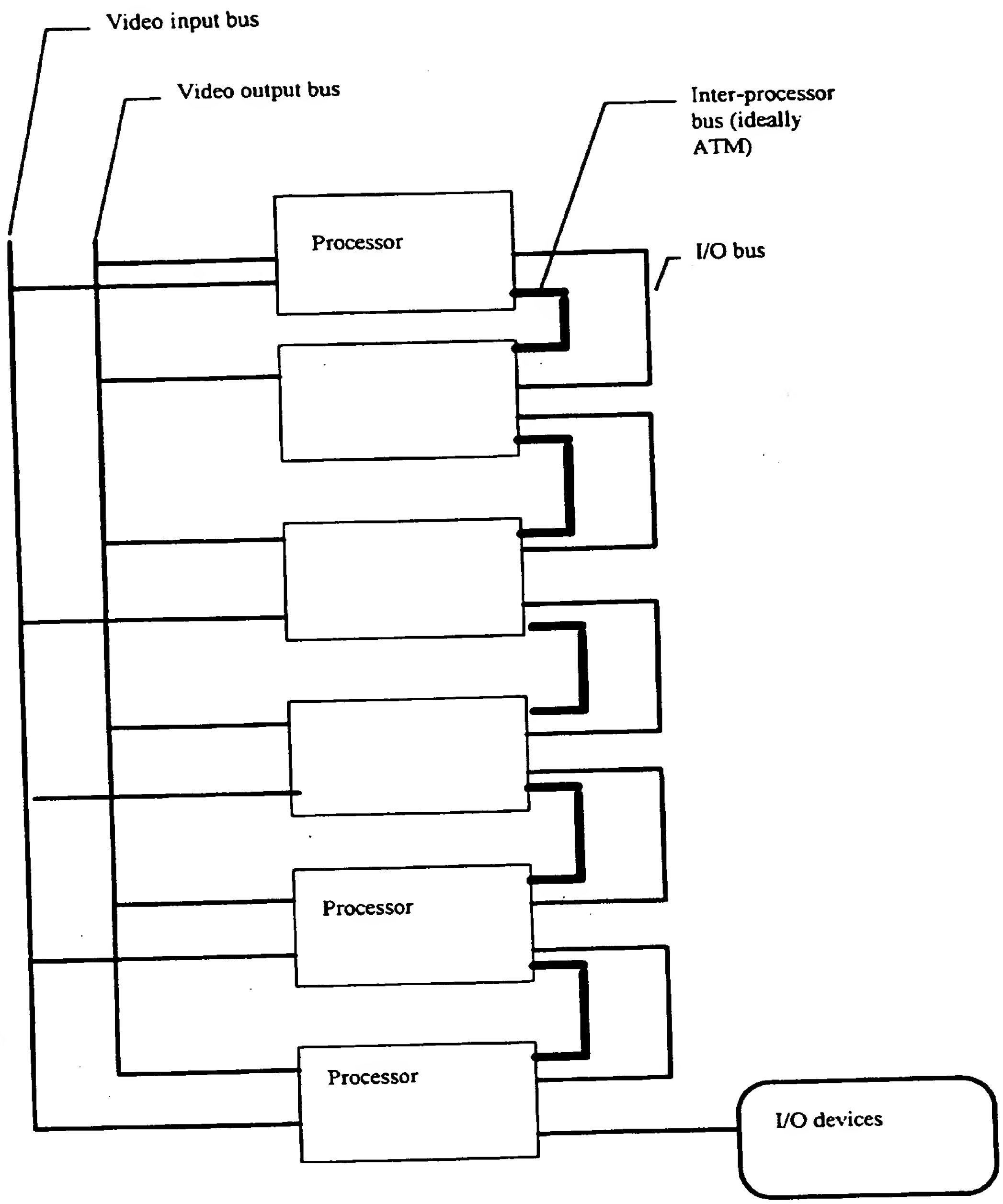


Figure 2 - Component Structure

Security

Consider the data path in Figure 3 The thick lines represent ATM circuits. Normally, a given virtual circuit (VC) will pass through the switch to the port programmed into it for that particular VC. Software controlled security is implemented by programming the switch to route the selected VC into that local processor. It is then a matter of implementation specific security rules as to the conditions under which the intercepted data will be re-inserted into the switch for delivery to the

child device. Note that compared to a conventional ATM switch, the re-injected data must have the same VC header as the incoming one. This avoids the need for the recipient device to be re-programmed to a different VC to the incoming one. The switch then comes transparent from a networking transparency perspective.

At low data rates, no hardware switch need be employed at all. All incoming cells may be inspected, filtered and routed under software control[3].

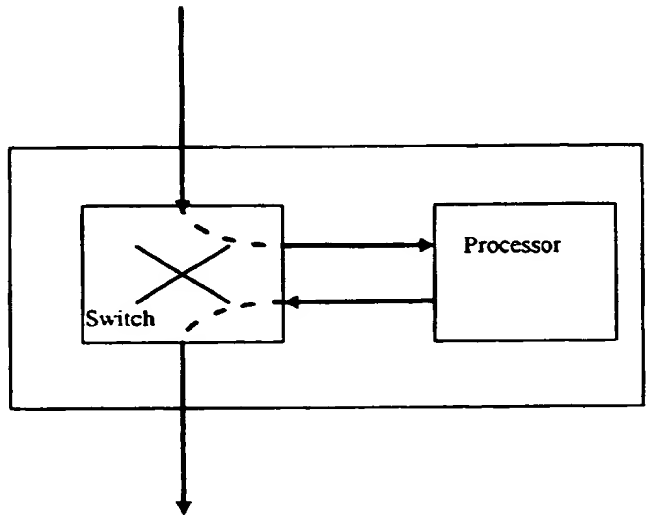


Figure 3 - Switched Data Flow

This structure permits the use of a very simple switch. Only those virtual circuits destined to be retained by the node need be registered. All others will be passed through unmodified.

Intelligent/Dumb Nodes

Figure 3 does require intelligence in a given node. Often, this may cause undue cost. Consequently, the structure must permit simple, dumb nodes containing just peripheral components that are unable to control the switch. This requirement is to permit traditional peripheral devices to be connected into the system without further intelligence[4].

The Asymmetric Security Processor (ASP)

This feature of the architecture sets it apart from all other architectures.

It is recognised that there is no such thing as a "secure system". There are merely different degrees of security. Greater security in general costs more and is more difficult to compromise. The IntAct bus provides a mechanism to tailor the degree and cost of security to the application.

The mechanism is The Asymmetric Security Processor. This permits differing degrees of security in each direction through the system and supports optional encryption schemes.

[3] This approach is intended to be used for very low bandwidth devices at 1Kbit/s or less.
[4] However, for the purposes of management and control, small microcontrollers are quite inexpensive and in most cases some intelligence can be afforded.

Level 1 - Virtual Circuit Security

The basic, ATM-like message passing provides basic security by routing all messages except those intended for a given node onto the next.

This level is suitable for embedded applications that would only get software upgrades via some protected service. Otherwise, rouge software could be used to re-program the node message switches and intercept messages. Even this though can be protected by judicious selection of the node sequence in the product:

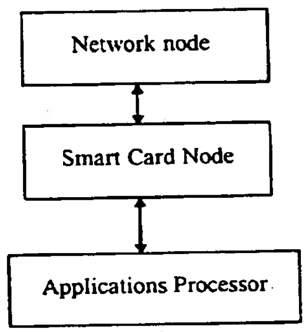


Figure 2 - Simple Secure Smart Card Application

The above example shows that a smart card support node may be used to prevent messages from reaching an applications processor. The latter is unable to intercept secure smart card messages such as those for biometrics or e-commerce simply because of its topological position.

Security provision is basic because a test instrument such as a logic state analyser could be used to record bus transactions. Then, an unauthorised node could be introduced into the productto intercept legitimate messages and then fabricate some form of system attack. Clearly, this vulnerability depends on having physical access to the nodes and so would not be a source of attack in say, an ATM machine or inside medical equipment.

Even if used inside a non-secure product such a Set-Top-Box, an attack would require physical access to the box and could only be used to defraud the user of that unit as a smart card would be used to confer conditional access. A set-top-box being used to access secure information would be an inappropriate tool for the job.

Level 2 - Simple Encryption

A logical exclusive-OR function is applied to each IntAct receive and transmit port. The exclusive-OR mask is programmed by the local processing in each node. The result is that all transitions on the IntAct bus do not have a apparently obvious meaning. All data valuse, including virtual circuit identifier, data type, data size and payload are encoded in this way.

The exclusive-OR mask may be modified from time to time via the IntAct Bud control protocol, making attack via a logic state analyser alone, worthless.

This approach imposes no bandwidth penalty on the system.

Level 3 - Autonomous Secure Conversation

This is the first of two system levels that attempts to foil intrusion by autonomous intelligence in the nodes.

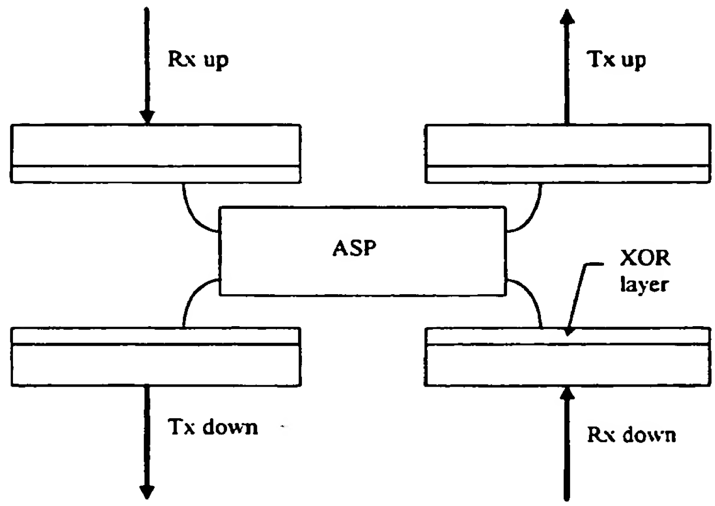


Figure 3 - ASP Location within a Node Switch

The exclusive-OR functions are controlled by the autonomous ASP. No attempt is made to uniquely encrypt every message that traverses the system as that would dramatically effect bandwidth. Instead, the ASP's in each node communicate with each other, exchanging public keys. Then, they encrypt and issue exclusive-OR masks to each other. Different masks must be used in the upstream and downstream ports of each node.

Only then will the ASP permit application traffic to traverse the network.

At intervals, the ASPs will re-establish communication and in a synchronised fashion, change the exclusive OR masks.

This approach has the following benefits:

- Mask generation and encryption is autonomous to ANY application intelligence and therefore is not subject to attack by it.

- Mask generation and encryption are calculated 'out-of-band' and therefore can be done with relatively simple, small, low cost microprocessors. This enables then to be embedded as macrocells within IntAct switch elements with only marginal effect on cost.

By changing the mask at intervals, any attack would have to be swift yet would only have access to part of the data.

Level 4 - Smart Card Validated Hardware

This approach augments Level 3:

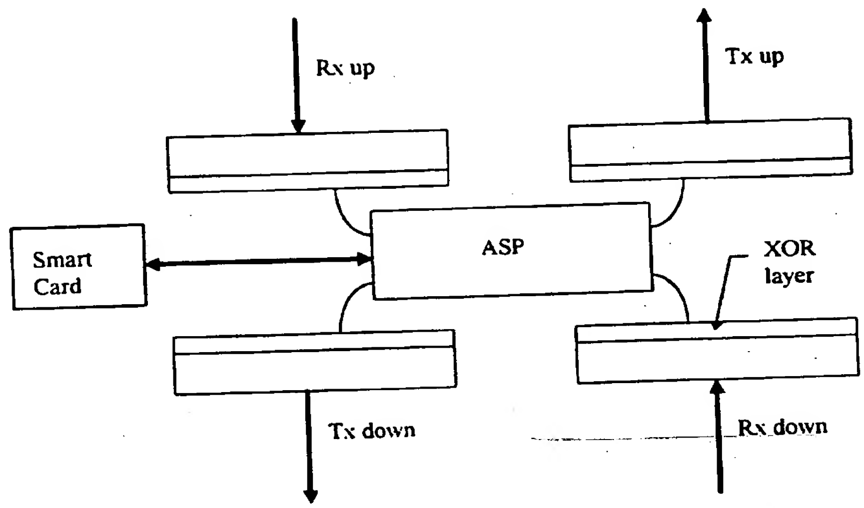


Figure 4 - Smart Card Augmentation

A smart card is used to provide a seed for the exclusive OR mask and to provide user authentication.

By removing the smart card, the node switch becomes completely inert, rendering the product useless. In a high security system, each node could be thus equipped. Removing the smart cards is virtually as effective as removing the entire product.

2

Appendix 1 - Sample Component Implementation

Component System Design

This section describes an implementation suitable for a wide variety of products in which a low cost, ATM-like inter-processor point-to-point network is provided. It specifies a system architecture that is compliant with the overall system topology and has features to permit near-term products to be constructed. In particular, CAN is provided as an object orientated extension of the IntAct architecture.

The key features are:

I^2C bus

For access to industry standard parts. This is intended for data transfers to 10Kbit/s to 100Kbit/s. It provides connection to motherboard programmable peripherals..

IntAct Module Bus

This is Amino's proprietary network designed to facilitate secure message filtering. It minimises the number of electrical interconnections on the system backplane.

Architectural and Application Connectors

The busses I^2C, IntAct and video, form the key components that every module must provide a connection to. Individual modules however are not obliged to utilise any given bus, save the I^2C. That is because this bus is used to identify modules inside a given enclosure.

Many applications will require additional connections that are of little use to others, such as video.. Mandating these connections will add unnecessary cost and complexity. Therefore two system connectors are defined. The first is the architectural connector containing the main busses. The other is an application connector that is implementation specific. Whilst the connector type is mandated, its presence and connections are not. There is no requirement for it ever to be used. This last statement has implications for mechanical stability. The application connector is not considered in the structural rigidity of the system.

Mechanical Layout

The following figures set out the principle mechanical details. Note that the module horizontal and vertical dimensions have two options, thus permitting four different PCB sizes. The key feature is that modules of different sizes may be mixed within a given enclosure. The Architectural Connector is the common feature to them all.

The constraint used to determine each of the key dimensions is as follows

Short Width	PCMCIA card plus socket with ejector mechanism
Long Width	PCMCIA card plus socket with ejector mechanism plus SODIMM socket width.
Short Height	208 pin PQFP part plus Architecture connector and allowance for trace routing.
Long Height	PCMCIA card plus socket with ejector mechanism plus Architecture Connector

Table 1 Card Dimension Criteria

Only the "Long Width variety have sufficient space to accommodate the application and architectural connectors.

The minimum module pitch is 12mm. Modules may have double sided assembly.

The formal definition of the Board Formats is as follows:

Format 1	45mm x 100mm
Format 2	80mm x 100mm
Format 3	45mm x 170mm
Format 4	80mm x 170mm

Table 2 - Board Formats

Architectural Bus Characteristics

I²C

This well known industry standard serial bus is resistively terminated on each IntAct module. However, this is primarily for test purposes. . Reliability is a more important goal than the highest possible performance. As more modules get connected to the card, resistive termination has the problem that driver current sink specifications may be exceeded. Consequently, a current source will be located on the motherboard. It will be set to just below the bus maximum sink current. This will ensure optimum speed and reliability.

The clock and data lines are both bidirectional and single ended. This bus is not suitable for extra-enclosure connection. It is really only suitable for single master use although its specification does allow multi-master.

IntAct Bus

This is a core technology for Amino and is covered in a separate document.

Ancillary Signals

A number of support services are supplied to all cards in the system.

- Main Power

5.0V, 3.3V and unregulated main power between 6V and 30V is supplied.

- System Reset

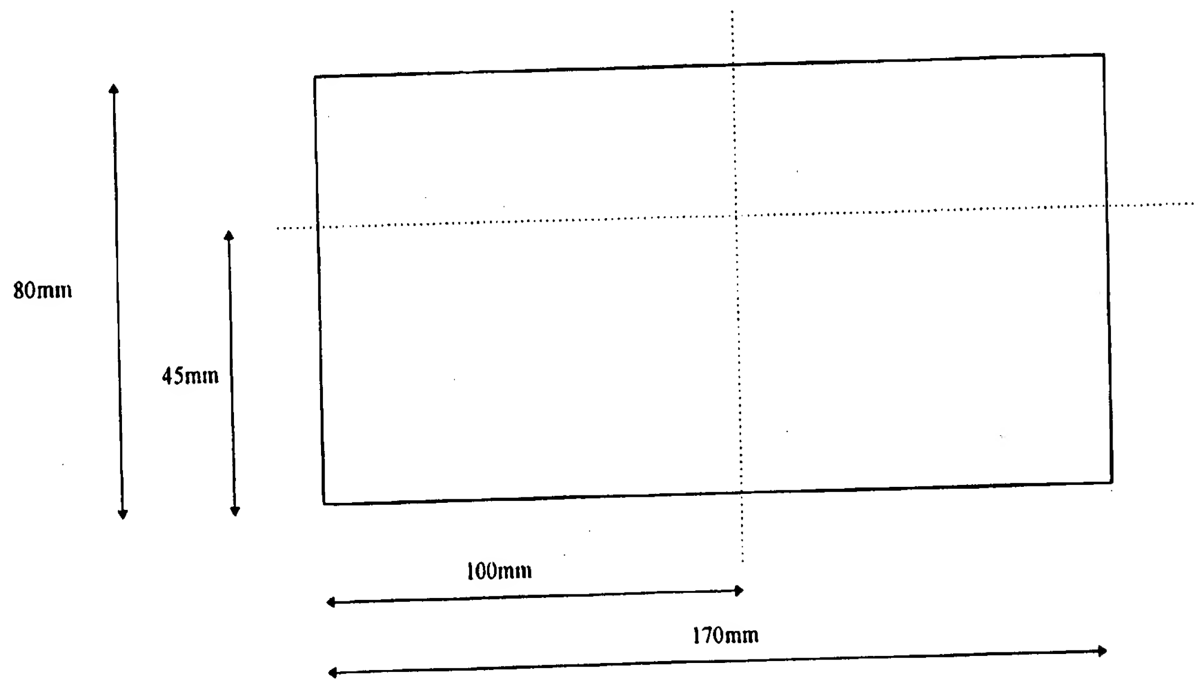
Used to re-start all modules. Certain high security modules may choose to disregard this signal.

- Power Fail

This signal indicates that main power will be removed anytime after the next 10ms.

- Battery back-up

An auxiliary power supply of between 1.0 and 4.8V



Note that the outline excludes the connectors but *may* include solder joints to the connector.

Amino Communications Ltd.

STRICTLY COMMERCALLY CONFIDENTIAL

Architectural Modules

Functional Specifications
23 January, 1998

1

Architectural Modules

Statements of Intent

Contents

The following sections describe each of the modules specified for compliance with the IntAct system architecture.

Module 1 - Format 1 CPU Module - MPC821, with FLASH ROM, DRAM and LCD video

Primarily intended as a main application processor module, this unit can provide all the elements of a portable Internet appliance, including LCD control, WAN/LAN networking, JAVA and connections to infra-red interfaces.

This highly compact module has the following features:

- MPC821 with software controlled clock speed.
- Single NAND FLASH ROM site.
- Single 32 pin NOR FLASH ROM site.
- EDO DRAM.
- 8 bit LCD panel video port.
- Battery backed real time clock.
- Soft Networking capability - Ethernet, PSTN, ATM and ISDN.
- I^2C, CAN and IntAct system busses.
- "Smart Media" card

Module 2 - Format 2 CPU Module - MPC821, with FLASH ROM, DRAM, PCMCIA and LCD video

Similar to MODULE 1, this unit additionally supports PCMCIA. It is very well suited to applications requiring external networking such as cellular data communications or access to SCSI peripherals.

- MPC821 with software controlled clock speed.
- Single NAND FLASH ROM site.
- Single 32 pin NOR FLASH ROM site
- EDO DRAM
- 8 bit LCD panel video port
- Battery backed real time clock
- Soft Networking capability - Ethernet, PSTN, ATM and ISDN.
- I^2C, CAN and IntAct system busses.
- PCMCIA Type 2 socket.
- Bi-directional Consumer I/R interface.

- ISO7816 / EMV Smart Card support.

Intended for medium performance PCMCIA equipped Internet devices, this module represents most of Internet client node in the volume of a cigarette packet. An additional module providing CRT video will normally be required.

RTC capacity is nominal and may be augmented by an external battery.

Smart Cards are supported via the Application Connector. Only "5V only" are supported. The interface requires an external Smart Card connector.

The module is also highly suitable as the main CPU module in dual (master - slave) CPU products.

Module 3 - Format 1 CPU Module - MPC823, with FLASH ROM, DRAM and video

This low-cost module can provide a complete set-top-box style Internet browser running JAVA. It can directly control a domestic television, requiring only an additional colour encoder.

It is also useful as a secure e-commerce engine, separating the user application from the e-commerce authentication.

- MPC823 with software controlled clock speed.
- Single NAND FLASH ROM site.
- Single 32 pin NOR FLASH ROM site
- EDO DRAM
- 8 bit LCD panel video port
- PAL/NTSC support (requires external PAL/NTSC encoder)
- Battery backed real time clock
- Soft Networking capability - Ethernet, PSTN, ATM and ISDN.
- I^2C, CAN and IntAct system busses.
- "Smart Flash" socket

Module 4 - Format 2 CPU Module - MPC823, with FLASH ROM, DRAM, PCMCIA and video

Similar to MODULE 3 this module has the benefit of PCMCIA and is well suited to providing a very flexible secure communications layer in Network Computers that are destined for a variety of network types.

Intended for medium performance PCMCIA equipped Internet devices, this module represents almost an entire Internet client node in the volume of a cigarette packet. Only the network physical layer and PAL/NTSC colour encoder need be added.

RTC capacity is nominal and may be augmented by an external battery.

Smart Cards are supported via the Application Connector. Only "5V only" are supported. The interface requires an external Smart Card connector.

This module is also suitable for use as the slave I/O processor in dual processor systems.

- MPC823 with software controlled clock speed.
- Single NAND FLASH ROM site.
- Single 32 pin NOR FLASH ROM site
- EDO DRAM
- 8 bit LCD panel video port
- PAL/NTSC support (requires external PAL/NTSC encoder)
- Battery backed real time clock
- Soft Networking capability - Ethernet, PSTN, ATM and ISDN.
- I^2C, CAN and IntAct system busses.
- Bi-directional Consumer I/R interface.
- ISO7816 / EMV Smart Card support.
- PCMCIA connector

Module 5 - CAN module with Object Orientated I/O and UART

This unique unit provides a variety of object orientated resources:

⇒ Remote serial port.

⇒ Optically isolated logic inputs and outputs.

⇒ Remote printer port

This module is intended for as an extension from higher performance IntAct nodes. Typical applications are:

⇒ RS232 - to CAN adapters.

⇒ General purpose digital I/O.

⇒ Parallel printer port connections for domestic and industrial use.

It provides these in a compact, low cost module that may be located up to 40m from the rest of the system.

- Single chip CPU
- 1M bit Nor FLASH ROM
- 32K byte SRAM
- UART
- Twelve bit plus hand-shaking parallel port.

Module 6 - Format 4 CPU Module - MPC603E, with FLASH ROM, DRAM and Digital Audio and MPEG video

This high performance module combines most of the features of a digital set-top box in a compact format suitable for high-end products. It requires an IntAct secure I/O module for e-commerce and general Smart Card support.

- 200MHz MPC603E
- Twin NAND FLASH ROM sites
- Single 32 pin NOR FLASH ROM site
- Synchronous DRAM
- PAL/NTSC support (requires external PAL/NTSC encoder)
- I^2C, CAN and IntAct system busses.
- Very High performance graphics engine
- Real time full motion ray tracing
- DVB transport demultiplexing
- AC3 digital audio
- MPEG2 decoding.
- 3D graphics support
- Software OSD.
- Optional integral JAVA byte code interpreter.

- Macromedia copy protection.

Module 7 - Smart Media Adapter

This simple IntAct module contains the interface electronics to connect Smart Media cards to IntAct systems with minimal cost.

- 5V and 3.3V card support
- Card insertion and removal alarms
- CAN or IntAct bus access
- Card ejector

Module 8 - Smart Card Adapter

This simple IntAct module contains the interface electronics to connect smart cards to IntAct systems with minimal cost.

- 5v only cards.
- Bull and ISO7816 positions.
- Card insertion and removal alarms
- CAN, I^2C or IntAct bus access
- Card ejector
- Keypad interface on Applications Connector
- Dedicated LCD controller interface.
- NAND FLASH program storage with ancillary static RAM.
- Consumer IR support

An Introduction To Amino Communication's

IntAct^{tm}

Secure Bus Technology

The Background

Modern small computer architecture has evolved over the last twenty years as a legacy of the Harvard and von Neumann architectures. These were concerned with how to put a computer together at all, rather than an attempt to envision a solution to application challenges at the end of the century.

The advent of the Internet, Smart Cards, e-commerce and digital video has challenged conventional architecture. What was originally designed for boolean and integer numerical operations is unwieldy and costly to deploy in what are essentially message passing systems.

Techniques used to offer wide, fast data busses are not intrinsically the best way to connect system functions together. Wide busses impose circuit board, manufacturing, product size and consequently, cost penalties on what need to be high performance, compact and low cost, flexible products.

Some Design Goals

Security

Whilst traditional computer designs go to considerable lengths to ensure security of data between applications, that security is compromised by the global nature of system busses.

Technological Transparency

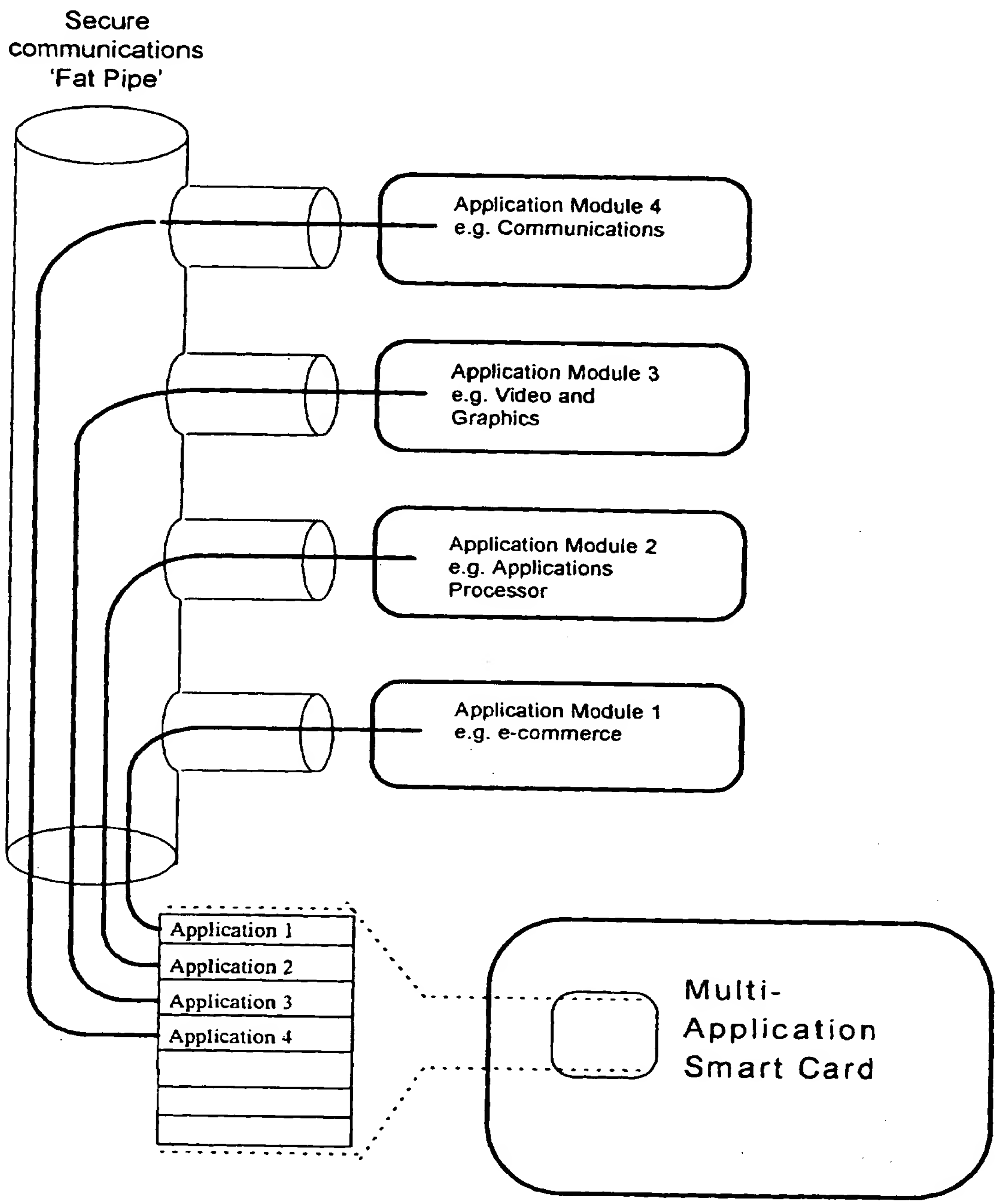
Crucially, many emerging applications and business models depend on security of data. A very large proportion would benefit substantially from the use of multi-application smart cards. Yet so many emerging products demand that at least two smart cards are in use. Consider domestic Digital Television products - one card is used to control say, conditional access and the other for purchases.

In fact, the plethora of products and services destined for Internet based distribution will be one of the driving forces behind business and consumer-orientated marketing and acceptanc of smart cards. Often the user will be unaware of the underlying technology. That is to be expected and indeed, welcomed.

The 'Technological Transparency' demands that products must be able to establish secure and reliable communications between product functions and smart card internal applications. The figure demonstrates the relationship between secure regions on a multi-application smart card and the functional modules within a product. The secure communications pipe serves all communications in the product, not just for a smart card. In the example given of a digital set-top-box, encrypted video and graphics will pass through the communications module and may be passed though to the video display module. That data must on no account be allowed near the applications processor in order to prevent illegal copying of the material. This permits an open product architecture which is still secure. *(NOTE - the figure demonstrates the data relationships - it does not convey how the system is actually constructed).*

Flexibility

The architecture imposes no constraints on which modules are connected to which others. It is up to the system designer to use the functions appropriate to the product.


Secure
communications
'Fat Pipe'
Application Module 4
e.g. Communications
Application Module 3
e.g. Video and
Graphics
Application Module 2
e.g. Applications
Processor
Application Module 1
e.g. e-commerce
Application 1
Application 2
Application 3
Application 4
Multi-
Application
Smart Card

Future-Proof

Modules may be arbitrarily replaced in the future. Conventional bus architectures demand all products must communicate no faster than the clock rate determined by the slowest device on the bus. IntAct^{tm} devices are an implementation of Intrinsic Technology Matching (ITM^{tm}) to ensure after-market module replacements are not constrained by the speed of earlier devices.

Low Cost

IntAct^{tm} is intrinsically a lower cost method of constructing systems than traditional architectures. As system complexity grows, IntAct^{tm} provides a scaleable and product foundation without burdening the system with ever increasing infrastructure costs. It is highly suitable for component-level integration with higher level functions.

Reduced Time to Market

IntAct^{tm} permits faster product introduction. For many products, the component modules will already be in production, thereby permitting just those addional, incremental parts to be engineered.

Highly Compact

The smallest IntAct^{tm} module format allows an entire, high performance Network Computer, Set-Top-Box, Industrial Controller or Vending Controller to be fitted within a volume similar to a pack of cigarettes.

Low Radio Emissions

IntAct^{tm} products are more readily compliant with world-wide EMC regulations than those with traditional architectures.

IntAct^{tm}

It is not really a bus at all. It is a series of module-to-module data relay mechanisms with many attributes of Asynchronous Transfer Mode (ATM) without the cost.

Subsystems within a product are constructed within modules that are interconnected by IntAct^{tm} data relay circuits. The sequence of modules is determined the desired data bandwidth and security between modules. The IntAct^{tm} specification sets down rules that the systems engineer will use when assigning the module interconnection sequence.

IntAct^{tm} is ideally suited to constructing products that use Ethernet, ATM, CAN or FireWire to connect with other devices. Cable modems, ADSL and PSTN dial-up devices are all readily connected to IntAct^{tm} modules.

Applications

The following list is merely representative. It is in no way exhaustive!

- Automotive
- Intelligent Consumer Appliances
- Internet Appliances
- Network Computers
- Embedded Web Servers
- Video-on-Demand
- Intelligent Cameras
- Navigation aids
- Communication devices
- Industrial

IntAct Bus Specification

Functional and Parametric Specifications
Revision 0.A.7 - 15 March 1998

Contents

Tables

Figures

Change History

Document re-structuring - Internal version 0.A introduced	MNG	6-3-98

NOTE for Revision 0.A: 6 March 1998

The removal of synchronous messages provided the opportunity to improve the programming model and make the message passing more efficient. In particular, the connection orientated signalling has been strengthened.

The overall bandwidth allocation between outbound messages and acknowledgements has been re-organised to be consistent with 32-bit processor data transfers between the IntAct interface and system memory. The net effect has also been to improve the inter-module software protocol efficiency by formalising the virtual circuit/virtual pipe analogy:

Consequently byte transactions have been dropped. The rationale is that the removal of synchronous transactions which were intended to support simple peripherals, typically accessed at low data rates has been replaced by funnelling in the local node. The byte wide mechanism had a maximum data transfer efficiency of 40%. The replacement format is 64% efficient. The 128 bit format is 87% efficient.

To allow data streams to be established with the IntAct equivalent to a permanent virtual circuit, a transmission or a reception is effected by reading the most significant data object (byte or word, as appropriate). Accessing the control or status registers will be completely passive.

This means a data source can send a data stream without the need to reset the header every time. The polled I/O / interrupt driven interface of the first implementation means reception must inspect each message type however the framework is in place for subsequent bus mastering.

1

IntAct Bus Specification

1. Functional Specification

The IntAct bus is a high performance serial communications bus designed as an inter-board datapath in high performance, low cost digital communications products.

Its key features are:

- Intrinsic datapath security
- Deterministic bandwidth allocation.
- Low cost of implementation.
- Low electromagnetic radiation (compared to wide parallel busses).
- ATM-like data structure.
- Intrinsic technology matching (ITM™)

The specification is based on the principle of predictable performance and controllable security and access to system components.

The structure is based on simple node-resident single buffer data switches with single comparator Virtual Circuit (VC) identification. Architecturally, it may be extended without limit within the structure shown. The version described in this document is suitable for up to 64 nodes.

2. Datapath Structure

The path is that of an ATM-like, full duplex route switched network. It's hierarchy is strictly regimented to ensure predictable performance.

At each node, the bandwidth requirements of all subservient nodes and local requirements must not exceed the availability of bandwidth above each one.

Latency is variable but will have a predictable maximum for each system implementation.

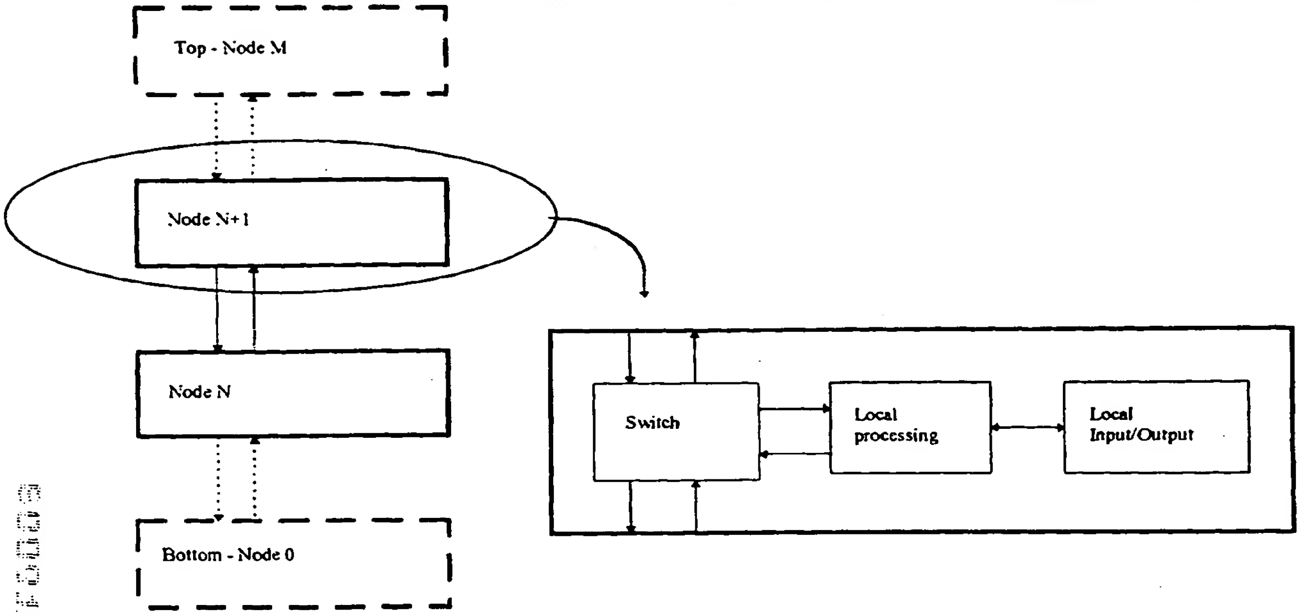


Figure 1 - Node Elements

Figure 1 shows the relationship between nodes.

- Data is passed up and down the chain.
- Any node may choose to pass data through, unmodified.
- Information is passed to specific destinations. The destinations may be logical or physical.
- Any node may initiate a transaction, up or down the hierarchy.
- Switches are only controlled by their local processors. However, that control may be over-ridden by local, secure cryptography controllers.

The switch in each node may be as complex as the application requires. However, the minimum functionality is as follows:

- All received messages destined for the local node are removed from the incoming stream.
- Received messages destined for nodes other than the local one are passed on.

3. Node Structure

Each node contains at least two of the three following elements:

- Local processing

- Local Input/output
- Local switch

3.1 Local Processing

Largely, this is some manipulation of data passing through the switch. It is distinct from manipulation of I/O functions although may use some input data from the I/O element. Equally, some of the data from the switch may be processed to be passed out of the system via the I/O.

Unlike traditional computer structures however, a significant portion of the bandwidth through the switch may be passed via the local processing. An example of this would be broadcast conditional access.

3.2 Local Input / Output

Module specific I/O processing is via the module's Application Specific connector or on-board. The latter should only be used if the board is not specified for use in multi-format enclosures.

3.3 Local Switch

This element gives the nodes their unique structure compared to traditional machine architectures. The local switch is described in 0 - 4. Local Switch.

4. Local Switch

The minimum specification for local switch functionality accommodates a very low cost implementation.

The switch supports full duplex traffic. Four data types in two data sizes are conveyed.

Table 1 lists the types and a brief description of their functions.

Data Type	Function
Cryptography Level 1: Non-secure.	Inter-node message passing between 'intelligent' processing elements. Nodes may only send data request messages or responses to earlier requests. They may only receive requests for data or the returned information for an earlier request. Each of these four activities is handled by a specific part of the switch. A traditional use of message passing of this type is to report interrupt requests and to carry network protocols.
Cryptography Level 2: Application processor mask encryption.	Transmission of pre-encoded messages between application processors to set up inter-node encryption masks. These are essentially special purpose Level 1 messages.
Cryptography Levels 3 and 4: Secure cryptography engine communications.	Similar to Levels 1 and 2, these messages are for communication between the cryptography processors of node switches.
Acknowledge	Message flow control.

Table 1 - IntAct Data Types

5. Data Types

5.1 Cryptography Level 1: Non-secure

Figure 2 - Asynchronous message between adjacent nodes, illustrates that this datapath involves bidirectional flow between two nodes. The source-to-destination flow is that of data: the destination-to-source is the acknowledge. Data flow between all other nodes is not involved or affected.

Figure 3 - Asynchronous message between separated nodes, illustrates that the 'inward' facing interfaces of the source and destination nodes and all interfaces of intermediate nodes will be involved, but not all at the same time. The timing sequence illustrates that the message is passed from the switch element of one node to the next.

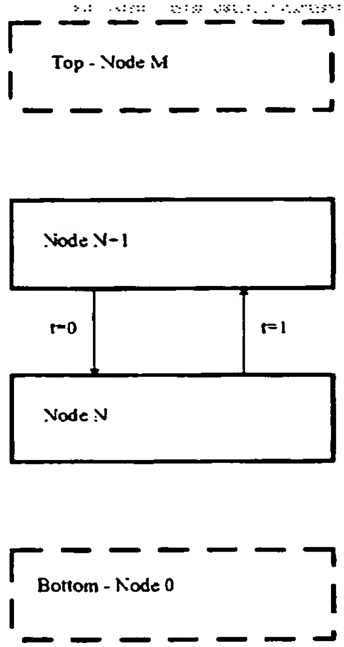


Figure 2 - Asynchronous message between adjacent nodes

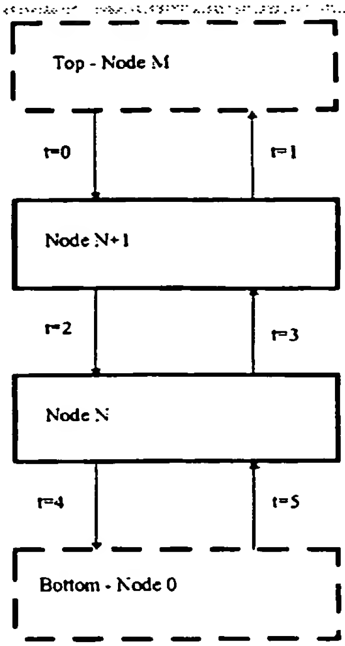


Figure 3 - Asynchronous message between separated nodes

5.1.1 Use of Acknowledge Response

This provides flow control. The minimal implementation is single buffered. More costly examples can be arbitrarily complex with deep FIFOs. In the latter cases, the data flow may be only infrequently interrupted. Single buffered examples passing data between separated nodes need to ensure that an incoming message has been at least passed through to the output stage of a given local switch element before the input stage is ready to accept another inward message. To this extent, the output stages taken together with the input stages provide double buffering.

5.2 Cryptography Level 2: Application processor mask encryption

Application processors communicate to set up the cryptography mask in their adjacent switches.

5.3 Cryptography Levels 3 and 4: Secure cryptography engine communications

This provides for switch internal and external cryptography engines to notify the encryption keys to their associated switches.

5.4 Acknowledge

This is the simplest part of the IntAct bus protocol.

These are only applied between adjacent nodes to regulate flow control. To avoid taking undue system bandwidth, only the code itself is carried. There is no route or payload data.

6. Security Levels

The various security levels use the data types to transport data between wither the applications processors or the Auxiliary Security Processors (ASPs) that may autonomously run the upstream/downstream network.

6.1 Level 1 - Virtual Circuit Security

The basic, ATM-like message passing provides basic security by routing all messages except those intended for a given node onto the next.

This level is suitable for embedded applications that would only get software upgrades via some protected service. Otherwise, rogue software could be used to re-program the node message switches and intercept messages. Even this though can be protected by judicious selection of the node sequence in the product:

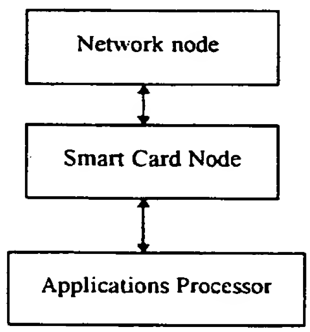


Figure 4 - Simple Secure Smart Card Application

The above example shows that a smart card support node may be used to prevent messages from reaching an applications processor. The latter is unable to intercept secure smart card messages such as those for biometrics or e-commerce simply because of its topological position.

Security provision is basic because a test instrument such as a logic state analyser could be used to record bus transactions. Then, an unauthorised node could be introduced into the product to intercept legitimate messages and then fabricate some form of system attack. Clearly, this vulnerability depends on having physical access to the nodes and so would not be a source of attack in say, an ATM machine or inside medical equipment.

Even if used inside a non-secure product such a Set-Top-Box, an attack would require physical access to the box and could only be used to defraud the user of that unit as a smart card would be used to confer conditional access. A set-top-box being used to access secure information would be an inappropriate tool for the job.

6.2 Level 2 - Simple Encryption

A logical exclusive-OR function is applied to each IntAct receive and transmit port. The exclusive-OR mask is programmed by the local processing in each node. The

result is that all transitions on the IntAct bus do not hav a apparently obvious meaning. All data values, including virtual circuit identifier, data type, data size and payload are encoded in this way. The data type is NOT encoded to permit the detection of un-encoded (type 1) or encoded (type 2,3 and 4) messages.

The exclusive-OR mask may be modified from time to time via the IntAct Bus control protocol, making attack via a logic state analyser alone, worthless.

This approach imposes no bandwidth penalty on the system.

6.3 Level 3 - Autonomous Secure Conversation

This is the first of two system levels that attempts to foil intrusion by autonomous intelligence in the nodes.

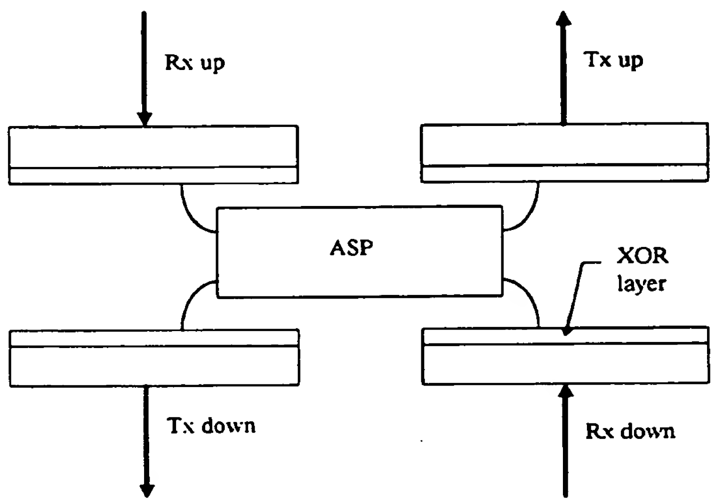


Figure 5 - ASP Location within a Node Switch

The exclusive-OR functions are controlled by the autonomous ASP. No attempt is made to uniquely encrypt every message that traverses the system as that would dramatically effect bandwidth. Instead, the ASP's in each node communicate with each other, exchanging public keys. Then, they encrypt and issue exclusive-OR masks to each other. Different masks must be used in the upstream and downstream ports of each node.

Only then will the ASP permit application traffic to traverse the network.

At intervals, the ASPs will re-establish communication and in a synchronised fashion, change the exclusive OR masks.

This approach has the following benefits:

- Mask generation and encryption is autonomous to ANY application intelligence and therefore is not subject to attack by it.

- Mask generation and encryption are calculated 'out-of-band' and therefore can be done with relatively simple, small, low cost microprocessors. This enables them to be embedded as macrocells within IntAct switch elements with only marginal effect on cost.

By changing the mask at intervals, any attack would have to be swift yet would only have access to part of the data.

6.4 Level 4 - Smart Card Validated Hardware

This approach augments Level 3:

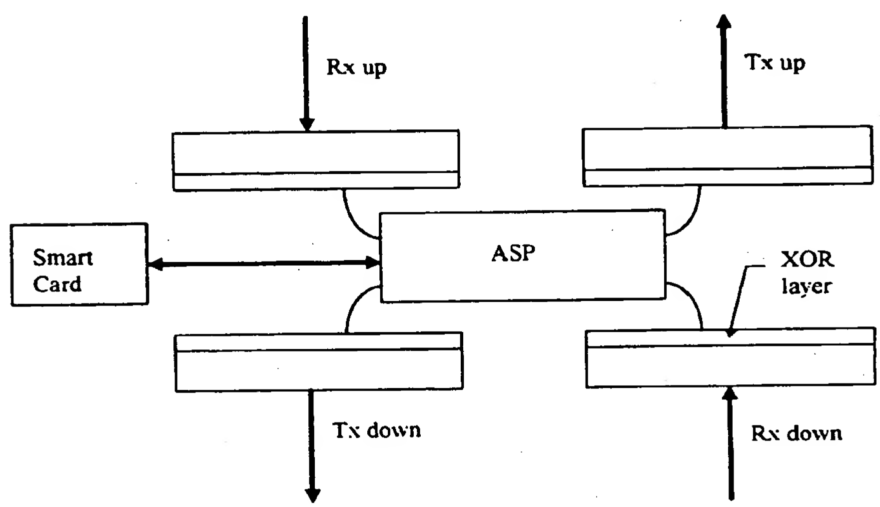


Figure 4 - Smart Card Augmentation

A smart card is used to provide a seed for the exclusive OR mask and to provide user authentication.

By removing the smart card, the node switch becomes completely inert, rendering the product useless. In a high security system, each node could be thus equipped. Removing the smart cards is virtually as effective as removing the entire product.

7. Message Formats

The structure of data passed in the messages follows.

The architectural definition allows either a bit-serial or nibble (four bit) serial implementation. This specification applies only to the bit serial form.

Left-most data is transmitted first.

Message Type (two bits)	Destination (six bits)
Data Size (two bits)	Source (six bits)
Payload (32 or 128 bits)	

Table 2 - Message Format

This arrangement simplifies the state machine logic as the bit counter and early termination can be processed during the following field.

The six bit route implies that this format supports up to 64 logical devices. This is deemed to be adequate for a physical system in which eight devices would be quite large.

The encodings for the message types are:

00	Level 1
01	Level 2
10	Level 3 and 4
11	Acknowledge

Table 3 - Message Format Encoding

The encodings for the data size bits are:

00	Zero (no payload)
01	Word (thirty-two bits)
10	Quad Word (128 bits)
11	(reserved)

Table 4 - Data Size Encodings

8. Switch Structure

8.1 Symmetric Blocks

The switch consists of two identical blocks. One sending data upstream and the other, downstream. The only intrinsic relationship between them is the automatic generation of Acknowledge responses. There is no data, route or switching relationship between them.

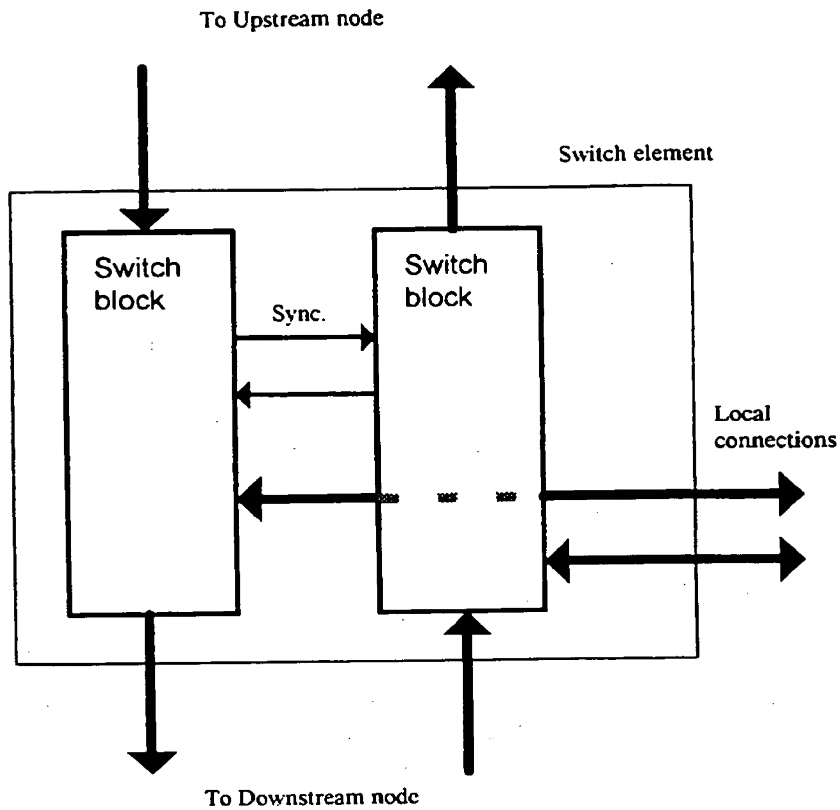


Figure 6 - Switch Element Structure

8.2 Block Structure

Each block contains of an input (receive) and an output (transmit) section. These operate under the control of a synchronising finite state machine. The state definitions are described in the document **"IntAct Signalling and Programming Model"**

8.2.1 Receive Section

The input section receives a message from the transmitter of another switch element, elsewhere in the system. It compares the route information with its own and if found to be equal, sends that information to the local processing or I/O in the same node. It only causes the other block to issue an Acknowledge response on its behalf when the local processing etc. accepts the received data.

If the data is to be passed on to another node, the receiver writes the message into the output buffer of the transmit section.

The receiver can be instructed to pass ALL messages to the local processing, irrespective of their destination. It is then the responsibility of the local processing to examine each message, pass it on, or issue another downstream message, related but not identical to the original message.

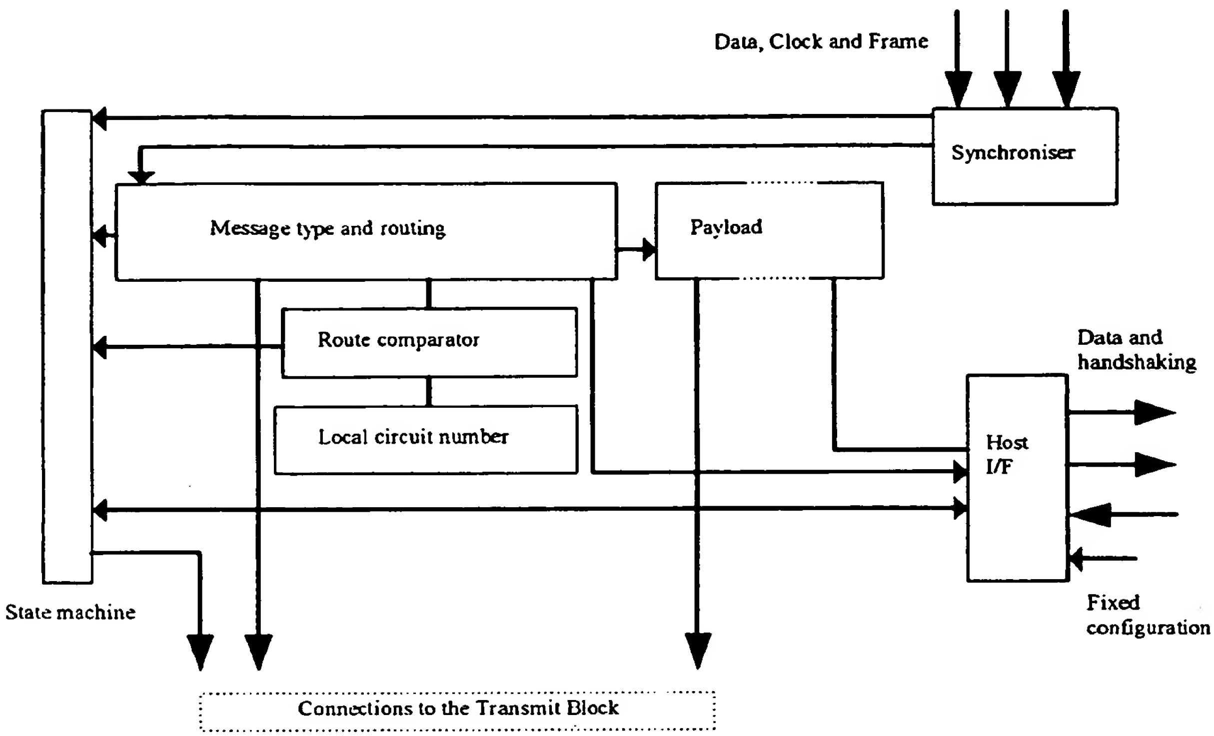


Figure 7 - Receiver Structure

8.2.2 Transmitter Section

As the transmitter can only send information from one source at a time, that is, either the receiver or the node's local processing, arbitration has to be applied to the transmitter buffer. Data is supplied to it either from the receive section or the local processing and/or I/O. To maintain bandwidth and latency rules, messages from the receive section take priority.

The simplest implementation is only single buffered and so local devices must be able to use wait states or use software polled READY status.

To indicate whether the transmitter is sending route or payload data, the message frame signal is asserted from the beginning of the message type to the end to the route information.

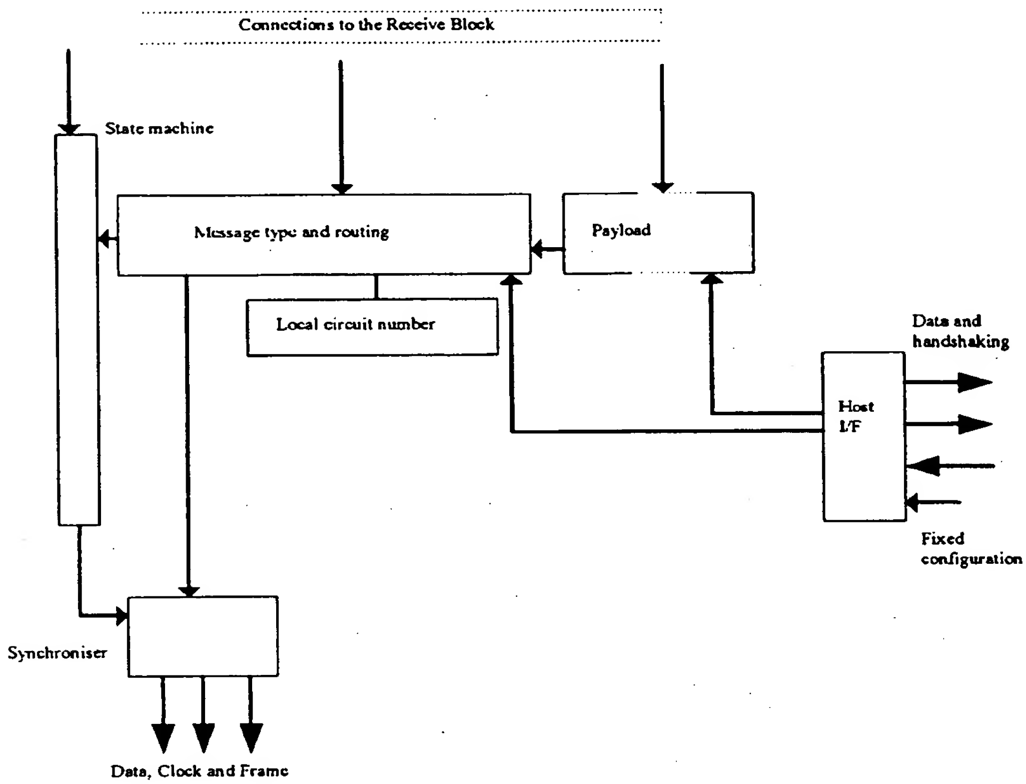


Figure 8 - Transmitter Structure

9. Message states

The inter-node signalling is described by the document **"IntAct Signalling and Programming Model"**. Each state is described at both the receiver and transmitter exchanging the information.

The purpose of these state machine descriptions is to facilitate the implementation of designs complying with this specification. No further interpretation is to be placed on these descriptions. In particular, unused states may or may not be used in later descriptions and implementations should not attempt to place any other interpretation in them.

10. Initialisation

10.1 Bottleneck Avoidance

Not all nodes will have the same capabilities. All must at least support asynchronous byte wide and synchronous byte wide transfers. All other features are optional. It is left to software routines to discover what other facilities are available. For example, local processing must establish that an thitry-two-bit only node does not exist between itself and a 128 bit node if 128 bit transfers are to be used. This could arise if a customer fitted a collection of nodes together in such a way. The design has to be robust enough to at least remain functional.

10.2 Identifier Resolution

Each node also has a type number and serial number. The same numbers are used as part of the CAN address if this too is available. Resolving chassis slot position to node identifiers is a software task that has minimal post-reset hardware assistance.

10.2.1 Chassis Position Resolution

In any given physical chassis, one node will be furthest up-stream and one will be furthest downstream.

The furthest up-stream node is deemed to be the chassis master for position resolution purposes.

Being the uppermost, it will NOT have an incoming clock signal on its outward facing receiver. During assertion of the system RESET signal, the presence or otherwise of the clock is used to determine whether or not a node is a master. The clock must be present for at least 256 cycles prior to the deassertion of reset. The fact that a node is a master will be indicated in a status register.

After reset, all IntAct nodes are configured with an address of zero.

Chassis position is determined from the master node onwards:

The master becomes device 0. It sends a down-stream message to node 1. The next node captures that message and increments the received circuit number, using the result as its own node address. The process continues, incrementing the virtual circuit numbers node by node. A given node MAY allocate itself more than one circuit number. This function may be done in hardware or by the local processing at a node.

11. Bus Clock Structure

The clock structure is unique and is an implementation of ITM™.

Transmitter and receiver pairs in adjacent nodes are connected so as to form asynchronous logic that generates a clock for the synchronous IntAct bus. This is illustrated in Figure 9.

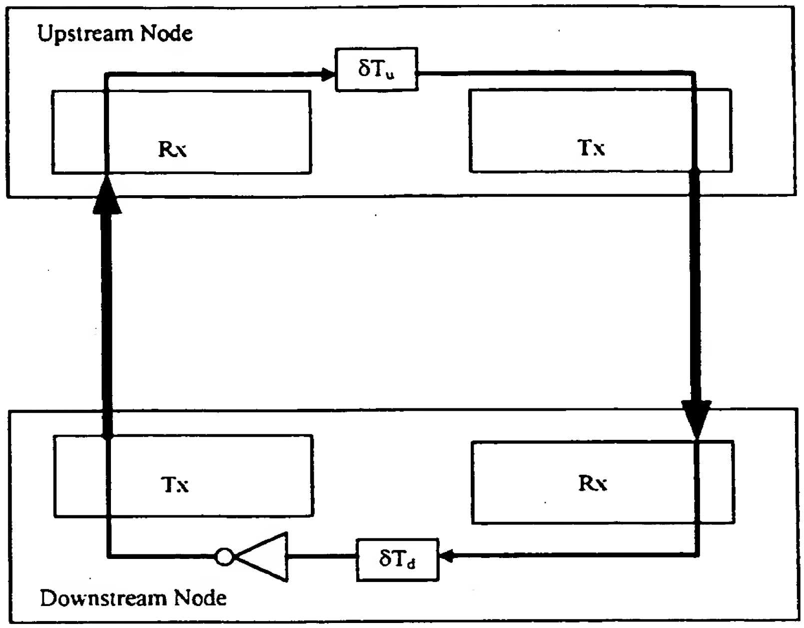


Figure 9 - Intrinsic Technology Matching: Clock Path

The clock path shown represents a loop with a gain of -1. That occurs at a frequency derived from the two delays, δT_u and δT_d of the upstream and downstream nodes respectively. It is a requirement that $\delta T_d > \delta T_u$. As it is a requirement of the IntAct bus that downstream bandwidth requirements are less than upstream capability, it is reasonable to assert that downstream technology may be slower than upstream. Even if not, there is no penalty by ensuring this condition.

The loop will oscillate at the frequency of approximately $2(\delta T_u + \delta T_d)$. It is a design requirement that the delay in either node is sufficient for a transmitter to send the next bit from its output shift register, or for its receiver to correctly receive and store an incoming bit.

Connecting transmitters and receivers in this way ensures that the complete loop will only transfer data at the rate of the slowest technology in the system. Subsequent developments deploying later, faster technology will automatically enjoy faster data transfer rate, limited only by the slowest adjacent nodes.

11.1 Minimum Delay Time

For the purposes of this specification, the minimum delay time is set at 2nS.

Overall data transfer rates will also be reduced by transmission line delays on circuit boards. These will usually be very small but the architecture ensures unreliability due to waveform degradation over transmission line dispersion is unlikely to occur as the loop speed will reduce with longer transmission lines.

12. Electrical Representation

Three signals are used to convey information:

- Clock
- Data
- Frame

Bit-synchronous timing allows the peak data rate between modules to be as high as possible without losing bandwidth due to pre-ambles for synchronisation. It also reduces the complexity of implementation and constraints such as phase locked loop layout constraints.

Subject to observance of set-up and hold times, there is no intrinsic restriction on clock speed. Independent transmit and receive blocks allow different speeds to be used on each, depending on whether or not messages are partially pipelined. Mid-message pipelining requires the same speed is used at both sides.

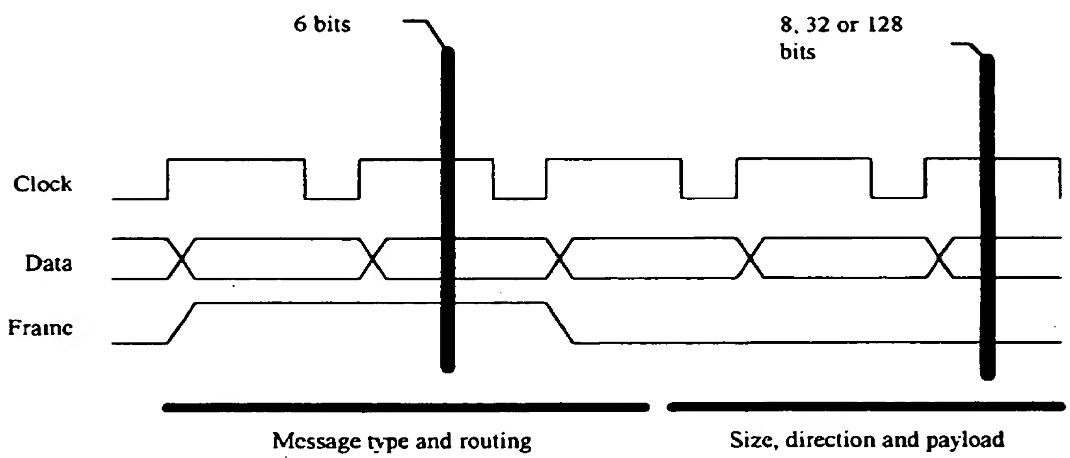


Figure 10 - Physical Layer Signaling

Signalling is based on 3.3V logic.

Receiver signals are all inputs. Transmitter signals are all outputs.

Setup time:	2ns minimum
Hold Time:	0ns
Bit Period:	6ns minimum

Table 5 - Bus Setup and Hold Times

13. Bus Signal Definitions

The following signals are a mandatory part of the interconnect specification:

UpTxData	Upstream Transmit Data	Bit serial data to an upstream node
UpTxClk	Upstream Transmit Clock	Outward path of the node's clock generation. It is the logical inverse of **UpRxClk**.
UpTxFr	Upstream Transmit Frame	Outward message frame indication to an upstream node.
UpRxData	Upstream Receive Data	Bit serial data from an upstream node.
UpRxClk	Upstream Receive Clock	Inward path of the node's clock generation. It is inverted to create **UpTxClk**.
UpRxFr	Upstream Receive Frame	Outward message frame indication from an upstream transmitter.
DnTxData	Downstream Transmit Data	Bit serial data to an Downstream node
DnTxClk	Downstream Transmit Clock	Outward path of the node's clock generation. It is the logical inverse of **DownRxClk**.
DnTxFr	Downstream Transmit Frame	Outward message frame indication to an Downstream node.
DnRxData	Downstream Receive Data	Bit serial data from an Downstream node.
DnRxClk	Downstream Receive Clock	Inward path of the node's clock generation. It is inverted to create **DownTxClk**.
DnRxFr	Downstream Receive Frame	Outward message frame indication from an Downstream transmitter.
nNodeP	Node present	This node is present in the system
nReset	Reset	Reset the node switch element. This may also be used

		to reset other functions.
vcc3	3.3v supply	Two connections with a total current carrying capacity of 2A are provided.
vcc5	5.0v supply	Two connections with a total current carrying capacity of 2A are provided.
DGND	Digital ground	Four connections with a total current carrying capacity of 4A are provided.
AGND	Analogue ground	Two connections with a total current carrying capacity of 0.5 A are provided.
CHASSIS	Zero potential well	No functional signals are to be returned through this signal. It is provided for EM shields to be connected to the system chassis. Four connections, two at each end of the overall connector are provided.

Table 6 - IntAct Bus Connector

Amino Communications Ltd.

IntAct and Legacy Systems

Legacy-Compatible systems
Revision 0.1, 18-Jan-98

Contents

1

IntAct and Legacy Systems

Constructing backward compatible systems with IntAct Modules.

Introduction

Amino's IntAct secure bus technology permits a wide range of secure products and services to be provided. The degree of security can be tailored to the end-product cost and to reflect the opportunity-cost of protecting the information or "content" delivered by the system.

IntAct modules are part of a larger architecture offering controlled bandwidth capability and secure data transfer within the scope of the architecture.

The Problem

The investment to design and manufacture entirely IntAct based domestic, commercial or industrial networks is considerable. The cost of companies replacing installed plant overnight would be huge and simply would not happen.

IntAct itself is however inexpensive. Computers, consumer products, networking and communications products constructed with IntAct could be less expensive than traditional architectures when called upon to provide the same degree of:

- Flexibility
- Security of data, access and electronic content
- Short time-to-market
- Low lifetime cost

These features make IntAct technology useful and desirable.

However, ubiquity of IntAct products will not happen overnight and they must co-exist with legacy technologies. IntAct technology and product development is based on a phased introduction of IntACt equipped products, leading to a licensing of the techlogy to other manufacturers.

The Marketing Aspects

Initially, the technology will merely be used to create "boxes" for a variety of purposes that have external characteristics similar to such products made with traditional architectures. Examples of these are Network Computers (NCs), digital Set-Top-Boxes (STBs), Automated Teller Machines (ATMs) and Automotive Data Terminals[1].

The competitive nature of these markets with the flexibility and low support costs they demand, is well suited to IntAct. Considered individually, any one of these markets represents a substantial business in its own right. Taken collectively, this reflects on the potential of IntAct to provide market pervasion.

IntAct can even be used to make secure IBM compatible Personal Computers (PCs), although these will cost more than traditional PCs. Such a development could be attractive for its ability to exploit existing applications software.

By establishing marketing partnerships with other organisations, Amino is able to offer its partners a better tool for the deployment of their own products and services. It is through such arrangements that the diversity of IntAct products will be made to grow and thus increase market penetration. It will be appreciated that Amino does not intend to make a feature of IntAct to the end-user, merely to use IntAct as a business development tool.

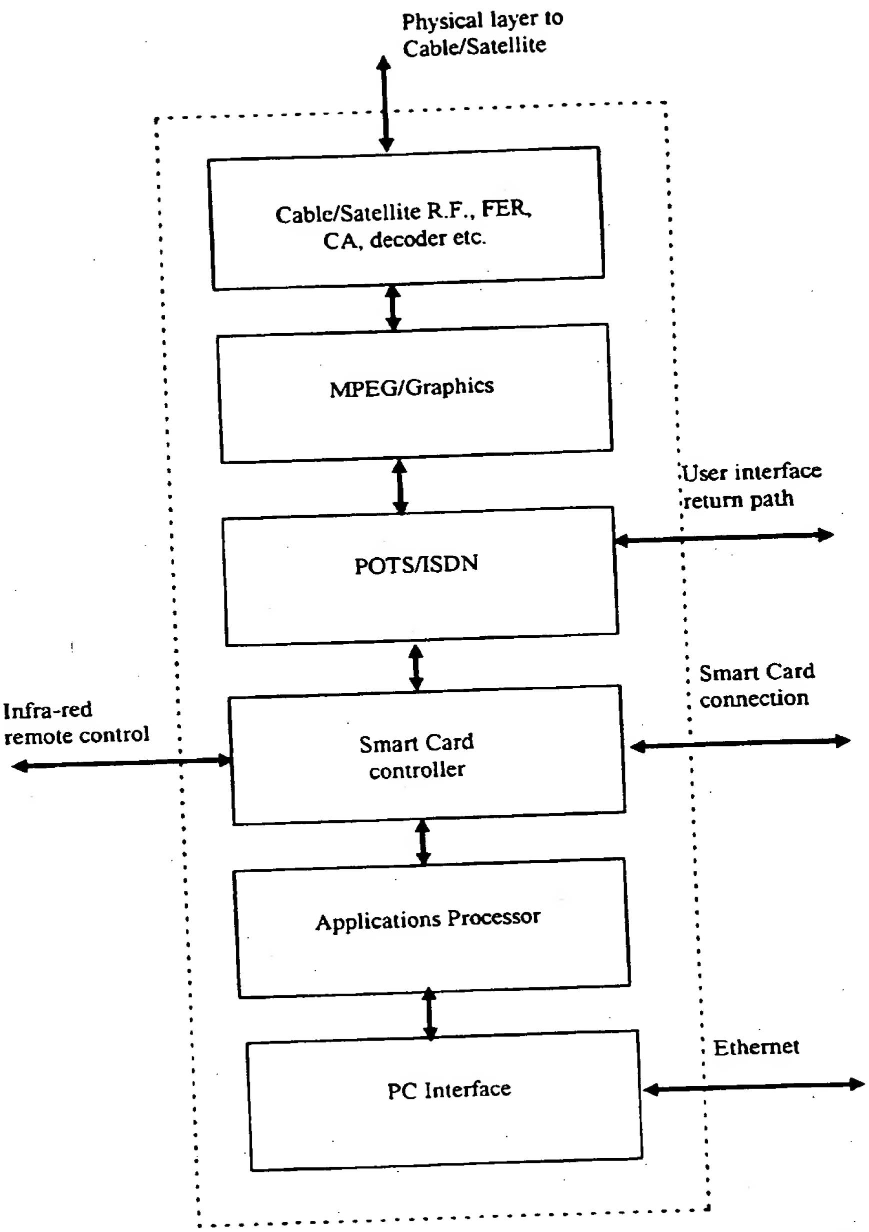
The earliest products will serve as a demonstration of the power and flexibility of the architecture. They will in turn be used as marketing tools to sell licenses to the technology[2].

The Technical Aspects

Here we concentrate on the architecture of specific products and how IntAct facilitates them.

[1] At the time of preparation of this document, Amino has an existing contract design and supply Automotive Data terminals and is negotiating to supply ATM machines and Asynchronous Transfer Mode connected Set Top Boxes.
[2] This is unlikely to succeed without some large an influential partners, investors and/or customers. This is why Amino is concentrating its investment efforts on larger, more influential organisations that bring with them much more than "just" money

Digital Set-Top-Box



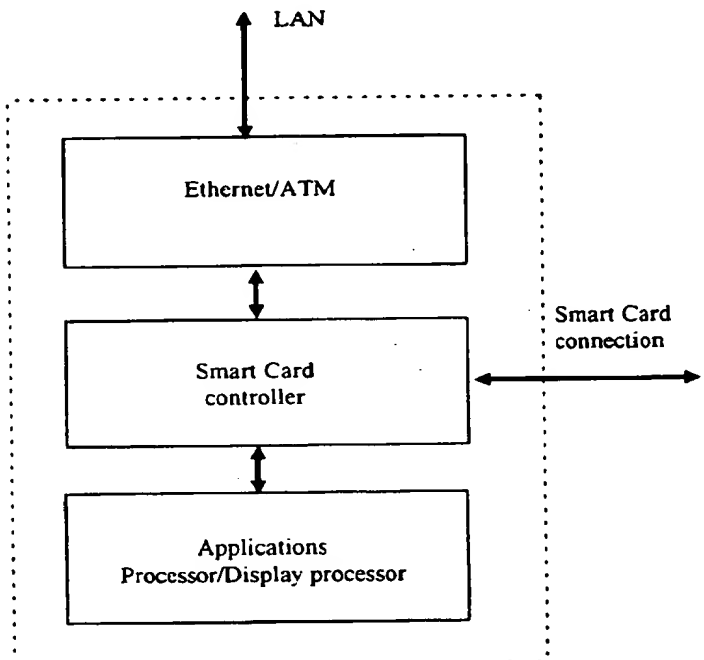
This example includes a POTS/ISDN modem as the user return path for satellite or cable telephony emergency backup.

An Ethernet interface allows the Box to behave as a router for the benefit of an external PC. As a router, its performance will be based on software packet switching and so would only be suitable for a restricted number of PC's on a connected LAN. Ethernet switch technology is well established and a "full service" IntAct/Ethernet module could be built.

Note that the smart card module lies between the applications processor (CPU) and the higher level networking and graphics. That allows the smart card module to intercede in any messages that the CPU might try to pass up the network. It also prevents the CPU from being able to directly access the smart card.

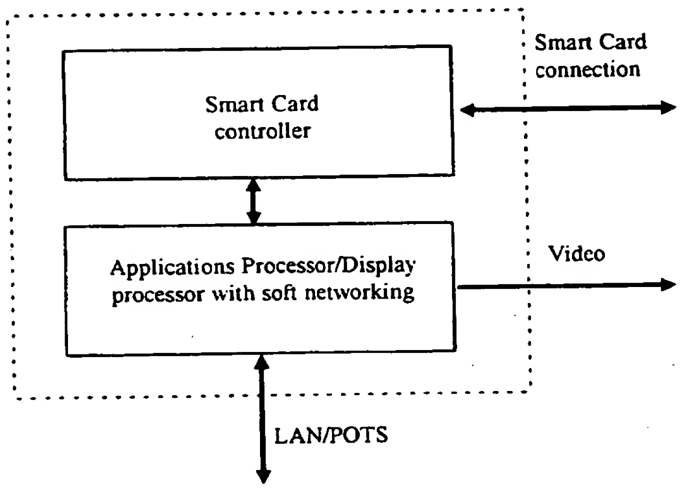
The POTS/ISDN interface would use TCP/IP for Internet communications, as would the cable interface. Digital video content would flow through from a cable interface to the MPEG decoder on a separate IntAct virtual circuit.

Network Computer



This is a simpler example. The smart card controller may not be needed if user authentication is performed some other way.

The following example is an even simpler network computer that uses an exiting module engineered by Amino that has on-chip networking. Note that it can also support POTS/Ethernet/ATM. The actual network in use would depend on the physical layer components in use.



The above computer could be simplified even further by omitting the smart card interface and using just the one module for the whole computer! Some motherboard parts would be needed for say, the POT Data Access Arrangement.

The above unit would also make for a very low cost personal Automated Teller Machine.

Web Phone

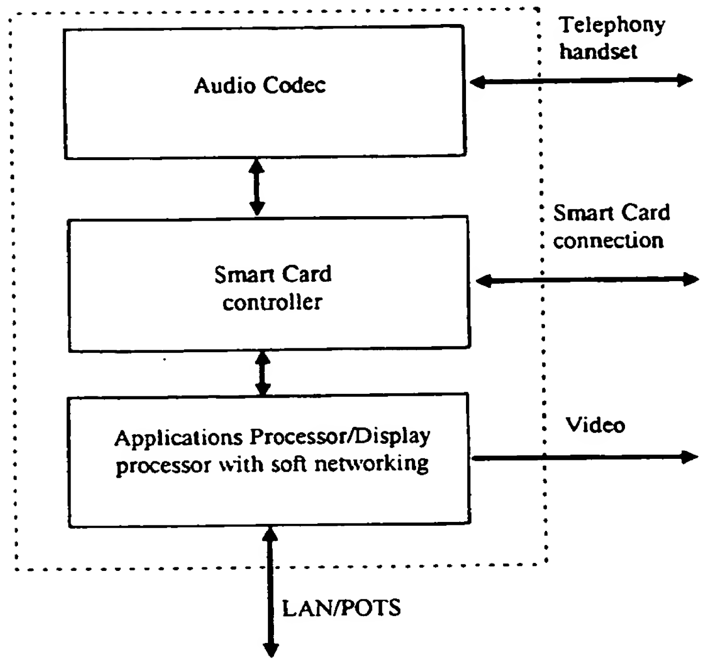
The following example would be suitable for voice over IP or merely as an augmentation of an analogue or ISDN telephone.

Again, a distinct smart card module is used to help ensure security of e-commerce or perhaps offer encryption of voice traffic passing through it, from/to the audio codec. The codec module could be very simple as IntAct supports synchronous or asynchronous traffic. A small microcontroller may be used merely to simplify configuration of what is a relatively 'dumb' device.

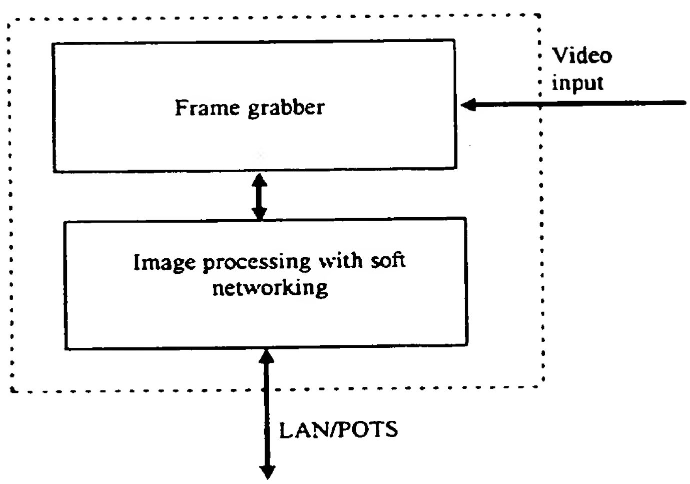
A LCD could be driven directly from the applications processor module (the existing design incorporates a Motorola MPC821 with on-chip LCD controller)

Bulk storage on a FLASH card is also supported on existing module designs[3].

[3] As at 19th February 1998. A module incorporation PCMCIA is currently specified for 2Q98 sampling.



Security Camera



This product uses the same module as employed as the applications processor in the Web Phone. All IntAct modules are fully re-programmable, reducing cost by increasing manufacturing volumes. Only an additional frame grabber module would be needed. Again TCP/IP would be used to communicate over the network.

Summary

These document has described a number of applications of varying complexity. They all operate with legacy networks and protocols; all are completely re-programmable to support up-loading of code improvements and new protocols.

In all cases, the fact that they employ IntAct internally to offer short-time-to-market and flexibility, is quite transparent to the user.